(2)

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

MULTIPARAMETER FORECASTING TECHNIQUES FOR
THE MARINE CORPS OFFICER
RATE GENERATOR

by

Charles J. Mehalic

September, 1990

Thesis Advisor: Robert R. Read

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b. OFFICE SYMBOL OR	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000	7b. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Including Security Classification)
MULTIPARAMETER FORECASTING TECHNIQUES FOR THE MARINE CORPS OFFICER RATE GENERATOR

12 PERSONAL AUTHOR(S)
MEHALIC, Charles J.

13 TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. Page Count
Master's Thesis	FROM TO	1990, September	112

16. SUPPLEMENTAL NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Forecast, Attrition Estimation, Harrison, Winters, Bayesian, Seasonality, Exponential smoothing, Skrinkage, Aggregation

19. ABSTRACT (Continue on reverse if necessary and identify by block number)
This thesis expands upon previous work in applying aggregation and shrinkage techniques to Marine Corps officer attrition rate estimators. Until now, estimation was based upon available annual data, failing to consider within year seasonality as a factor. Exploring modern short-term forecasting techniques which include a seasonal factor, this research applies seasonality on a quarterly basis with conversion flexibility to any desired cycle.

We introduce and compare two models: the Harrison-Stevens Multi-State Bayesian model and the Winters Three-Parameter Exponential Smoothing model. Both methods provide capable forecasting and demonstrate the necessity of including seasonality. The Harrison-Stevens approach has the advantage of providing a posterior distribution rather than a point estimate, and proves to be the superior model when forecasting beyond one period.

20 DISTRIBUTION AVAILABILTIY OF ABSTRACT X UNCLASSIFIED/UNLIMITED SAME AS RPT. DTIC	21a. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Robert R. Read	22b. TELEPHONE (Include Area Code) (408)646-2382 22c. OFFICE SYMBOL OR/Re

Multiparameter Forecasting Techniques for the Marine Corps Officer Rate Generator

by

Charles J. Mehalic
Major, United States Marine Corps
B.S., United States Naval Academy, 1976
M.S., University of Southern California, 1981

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 1990

Author: ______________________
Charles J. Mehalic

Approved by: ______________________
Robert R. Read, Thesis Advisor

____
Lyn R. Whitaker, Second Reader

____
Peter Purdue, Chairman
Department of Operations Research

ABSTRACT

This thesis expands upon previous work in applying aggregation and shrinkage techniques to Marine Corps officer attrition rate estimators. Until now, estimation was based upon available annual data, failing to consider within year seasonality as a factor. Exploring modern short-term forecasting techniques which include a seasonal factor, this research applies seasonality on a quarterly basis with conversion flexibility to any desired cycle.

We introduce and compare two models: the Harrison-Stevens Multi-State Bayesian model and the Winters Three-Parameter Exponential Smoothing model. Both methods provide capable forecasting and demonstrate the necessity of including seasonality. The Harrison-Stevens approach has the advantage of providing a posterior distribution rather than a point estimate, and proves to be the superior model when forecasting beyond one period.

THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

TABLE OF CONTENTS

I. INTRODUCTION

A. GENERAL

This thesis investigates the use of modern short term forecasting within the framework established to build an attrition rate generator for a large scale manpower flow model; specifically the officer force of the U.S. Marine Corps.

The Marine Corps is continuing the process of automating their manpower planning, programming and budgeting processes with the recently developed and highly organized Officer Planning and Utilization System (OPUS). As this centrally organized system evolves, more efficient methods of anticipating personnel attrition from the Corps are needed.

Attrition includes those leaving the service through retirement, resignation, discharge, disability or similar reasons. The Navy Personnel Research and Development Center (NPRDC), San Diego, California, recently terminated efforts in forecasting attrition through the Marine Corps Officer Rate Projector (MCORP). Decision Systems Associates, Inc. (DSAI), of Rockville, Maryland, has been granted the contract for future implementation of their forecasting Officer Rate Generator (ORG).

Accurate forecasting of officer losses is extremely important to the manpower planner. In military manpower systems, most personnel flows are initiated by the creation of vacancies within the system. Losses in the paygrade hierarchy trigger promotions from lower grades. Vacancies generate a need for new accessions to replenish the force. The lead time in this process is great, thus quality forecasts are essential. Underestimates of losses lead to too few accessions, erroneous budget projections, and untimely readiness problems. Overestimates of losses can cause excess accessions, promotion delays, underutilized personnel and increased costs. The problem is compounded in that most accessions begin at the lowest pay grade of Second Lieutenant and slowly work their way up to the highest ranks over a period of many years.

The present attrition rate generator calculates empirical attrition rates using historical data with user-defined weights and threshold parameters (Seigel, 1983). This subjective input makes the current generator susceptible to unintentional misuse. In support of the ORG, Professor Robert R. Read of the Naval Postgraduate School has been working on a number of modern techniques applied to the problem of estimating attrition rates for the numerous cells that appear in manpower planning models. Special attention has been given to the "small cell" problem; i.e., officer categories that normally contain a few personnel. These cells are numerous,

and historical empirical rates for them are generally unstable due to sporatic data. In addressing this problem, the contract granted to DSAI specifically requires the implementation of shrinkage techniques developed by Professor Read and Major J. Misiewicz (Misiewicz, 1989). Their solution to the "small cell" problem is explained in Chapter III.

Due to data constraints, Misiewicz' thesis application is based upon annual data and cannot provide for attrition variability due to different seasonal periods. It must be considered that attrition rates may be seasonally dependent. In our study, the readily available data requires that we approach officer attrition using the calendar year broken down into three-month periods (quarters) and analyze various seasonal forecasting techniques. The more refined objective of the Marine Corps is to develop the capability to forecast attrition on a monthly basis with projections to any future month desired.

B. BACKGROUND

Eight Master's theses have been produced by this project. Each has made important contributions to the understanding of the problems associated with the "small cell".

Major D. Tucker provided detailed background into the Marine Corps officer structure and the manpower planning process (Tucker, 1985). He provided basic attrition rate theory and calculated attrition rates in several formats. He

tested three estimation schemes: maximum likelihood, James-Stein, and minimax for a few selected paygrades and military occupational specialties (MOS). His results strongly support use of James-Stein estimation of attrition rates. Minimax was discarded as being too conservative for small cell use.

Major J. Robinson introduced the Efron-Morris limited translation shrinkage alternative to augment the James-Stein estimator (Robinson, 1986). He performed a more thorough validation using both original and transformed scale, and while confirming Tucker's results, he could not provide consistent stable estimates for small cells.

Captain C. Dickinson continued the application of shrinkage methods to estimating officer attrition rates (Dickinson, 1988). He applied the previously used methods and an empirical Bayes estimator to a new and refined data base which recorded "man-quarters" rather than "fiscal" data. This approach was competitive with previous methods but instability remained.

The next three studies used ad hoc methods to deal with the idea of cell aggregation. When applying shrinkage methods, aggregation of cells with low personnel inventory into sets of cells with larger inventory is required. It was believed that a mathematical approach to this question would give relief to the instability problem encountered by others. The objective is to use cells which demonstrate similar attrition behavior.

The first attempt to treat the aggregation problem was H. Amin Elseramegy (Amin Elseragemy, 1985). His use of the Classification and Regression Tree (CART) program in forming aggregates of cells exhibiting homogeneity of behavior in attrition proved difficult. Efforts to learn the system, computer memory space on the IBM 3033, and random partitioning of data in this "top-down" system was limiting. The resulting aggregations were awkward and generally unusable.

Substantial progress in the aggregation problem was made by R. Larsen (Larsen, 1987). Using a second, more refined data base, he applied a hierarchical clustering algorithm and exposed the relative importance of some special MOS cells and years commissioned service (YCS) intervals. The separation of the aviation community into several groups explained much of the instability encountered in earlier studies. Larsen's work provided the framework for the cell aggregation algorithm developed by Misiewicz.

D. Hogan turned his attention to alternative methods for attrition estimation (Hogan, 1986). Believing the existence of a time series effect, Hogan explored an exponential smoothing technique. This technique provided a way to update estimates yearly with the passage of time through a weighted smoothing constant, α. The results indicated that exponential smoothing gave relief to the problem of estimating rates using large time lags but with inconsistent results. Further study

into exponential smoothing is contained in this thesis and discussed in Chapter V.

A logistic regression alternative was explored by N. Yacin (Yacin, 1987). This study provided some quantitative results regarding similar attrition behavior with respect to years commissioned service (YCS).

J. Misiewicz built upon the results of these previous studies. Initially he integrated two efforts:

- the aggregation of cells into groups that exhibit homogeneity of attrition behavior, and
- the development of "shrinkage" estimation techniques for use in the individual groups.

A heuristic algorithm was developed and tested to treat the aggregation problem with empirical Bayes methods used to serve the multi-cell estimation requirements needed to preserve fidelity. In essence, it is a modification of the Larsen procedure. His results indicate stability in estimating attrition rates for low inventory cells but presented no clear favorite among six estimation methods. DSAI is presently integrating the Larsen-Misiewicz small cell algorithm into the ORG using the transformed scale, time dependent variance method. Chapter III is an amplification of Misiewicz' efforts. (Misiewicz, 1989)

C. OBJECTIVE

This thesis continues ongoing research in the development of the Officer Attrition Rate Generator for the U.S. Marine Corps. Successful effort has been given to refining an aggregation and shrinkage technique for handling the inherent problem of forecasting in a small cell environment. General stability is achieved through MOS and YCS grouping which replaces the earlier ad hoc methods that led to historical instability concerns.

The objective of our work is to tune the aggregation and shrinkage algorithm developed by Major John Misiewicz, then focus on estimation techniques which consider seasonality forecasting factors. Two specific techniques are developed. They are based upon:

- Winters Three Parameter Seasonal Exponential Smoothing Model.
- Multi-parameter Estimation and Forecasting using the "P. J. Harrison and C. F. Stevens" approach in a finite-state model.

The algorithm developed by Misiewicz is modified to view forecasting from a "quarterly" perspective. A new data tape is provided by MIIS, Headquarters, U.S. Marine Corps to assist in this work, providing twelve years of data rather than ten years as used in previous work. Most of Major Misiewicz's work with small cell aggregation and expansion techniques will remain intact, with moderate modifications of the expansion parameters.

The Winters Seasonal Exponential Smoothing technique is appropriate for seasonal time series data. It uses three separate smoothing constants to describe the level of the series, a linear trend, and a multiplicative seasonal factor.

Harrison and Stevens describe a new approach to short-term forecasting, based upon Bayesian principles in conjunction with a multi-state data generating model. The various states correspond to the occurrences of transient errors and step changes in trend and slope. The basis for this method is founded upon what is commonly referred to as "Kalman Filters" and should provide the following advantages:

- Recognition and responsiveness to transient errors and sudden changes in trend and slope.
- Increased sensitivity when true trend, slope and step changes occur.
- A joint distribution rather than a single-figure forecast.
- Known values for seasonality which can change with additional data sets.

II. DATA BASE

A. GENERAL

Previously, the works of Larsen and Misiewicz used a refined data base compiled by NPRDC. It contained ten years of inventory and attrition data from 1977-1986. The inventory data provides annual totals of officer inventory in units of man-quarters and was obtained from the Headquarters Master File (HMF). The attrition data was accumulated in man-years from the Quarterly Statistical Transaction File (STATS). To be used together, the inventory data was divided by four in order to convert to an annual (but not integer-valued) figure.

With NPRDC no longer on contract and DSAI not expecting to obtain a usable monthly data base until September 1990, we allied with MIIS, Headquarters, U.S. Marine Corps, Washington D.C., for the preparation of a new data base. The NPRDC data base contained annual attrition and inventory data by YCS, paygrade, MOS, sex, commissioning source, education level and service component. With multiple data base problems in conjunction with limited time and resource, this degree of detail was determined to be too ambitious.

Our objective was to obtain a central inventory and attrition value for each quarter on cells defined by MOS, paygrade, YCS, and service component. Unfortunately, the only

available inventory measures are instantaneous. At best, the data base includes a snap-shot of the inventory on the final day of the quarter and a tally of attrition over each cell. Since attritions are not included in the inventory value, the question arises as to what value to use for quarterly inventory; the snap-shot value alone, or the snap-shot value plus the quarterly attritions. Because many Marines transition in and out of a cell during a period with only a few being attritions, our computations are based upon the end-of-period value. Numerous problems were encountered with this data base, most notably lost records and significant attrition rate outliers. To deal with these problems, it was necessary for us to develop an outlier identification and replacement system. Other discrepancies noticed include:

- Some negative value entries are listed for YCS (we assume and change these to the equivalent positive value).
- In many senior officer records secondary MOS rather than primary MOS are listed (without primary MOS information, these records are unfortunately ignored).
- Listing of nonexistent MOS (we also ignore these records).

These discrepancies were dealt with individually.

B. FILE DEVELOPMENT

Appendix A displays a sample of the data found on the tape provided by Headquarters, U.S. Marine Corps. The output displays a single entry with the corresponding description of

what each field represents. Each entry is generally described as a count of attrition and inventory for a given cell defined by period, paygrade, YCS, MOS, and service component. The coding of the data base is identical to that in previous work with the following exceptions:

- The data base contains only paygrades O1 (Second Lieutenant) through O5 (Lieutenant Colonel).
- Some paygrades are followed by the code "E" to signify prior enlisted Limited Duty Officer (LDO). As in prior studies, we will limit our work to unrestricted officers, therefore, these data entries will be ignored.
- YCS is truncated to an integer rather than rounded as in prior work. This is compensated for in the FORTRAN program MCFIND which develops our data base.
- Actual primary MOS designations are used rather than substitute codes, e.g., 0302 is listed for basic infantry officer rather than 013 as found in prior work.
- Service component is given a code of "1" to represent an officer with a regular commission, and "2" to represent an officer with a reserve commission.
- The period is designated using the last two digits of the year and the two digits of the final month of the quarter (e.g., 7903 represents the first quarter of 1979, 8612 represents the fourth quarter of 1986).

In order to make the data base usable, a FORTRAN program named MCFIND (Appendix B) was written which reads all records in the main data base and develops a source data base (example in Appendix A) for our actual use in the forecasting algorithm. As in prior work, only unrestricted MOS fields are considered; therefore, many records are eliminated from consideration. In addition to correcting the YCS entries this

program selects and compiles the appropriate source data base by reading the records of only the applicable paygrades (O1 through O5) and MOS (those listed in Table 1 of Appendix C).

C. OUTLIER IDENTIFICATION AND REPLACEMENT

1. Policy Requirement

The existence of inventory outliers in the data base is evident and is critical. The outliers usually relate to system undercounts. Discussion with MIIS personnel at Headquarters, U.S. Marine Corps yields the recognition of periods of data omissions for reasons which are unexplained. In some cases the undercount in small while in certain cases in which a significant inventory is known to exist, few or none were recorded. In order to use the data base provided, it is necessary to develop a policy to identify and replace these outliers.

2. Outlier Identification

Using periodic (quarterly) data, cross-classified by paygrade and MOS group, we apply our outlier identification procedures to the inventory values. This particular aggregated classification of the data base is created using our FORTRAN program MCMATX (Appendix B). Our purpose for using this macro cross-classification is to provide for a larger and more stable basis of outlier identification. The outlier identification procedure is simple. For each paygrade in a given MOS group, we find the inventory median over all

periods; an outlier is identified as being any period with an inventory deviating from the median by more than two times the interquartile range. This paygrade/MOS group/period combination is then tagged as an outlier. This tag additionally identifies each corresponding small cell (those further characterized by YCS, specific MOS and service component) outlier.

3. Outlier Replacement

Having identified the outlier cells, our first approach is to replace the outlier cell with the median inventory (target inventory) taken over all cells having the same paygrade/MOS group over the three corresponding periods both previous to and following the observed outlier. This aggregate inventory level can then be extended to the smaller classification including YCS (paygrade and MOS group is further refined) and service component. While this approach is sound given an adequate sample size, such a sample did not always exist. Specifically, in the 12th period (the fourth calendar quarter of 1980) the inventory recorded for Captains of all MOS is ten or fewer.

An alternative replacement method is implemented which replaces the tagged outlier cells with the mean inventory and attrition values taken over the preceding four periods and succeeding four. Because there are no four succeeding periods in the case of period 45, three are used instead. Though

biased, this method is simple and is only applied in a few rare incidents as discussed later.

4. Application

The identification of inventory outliers is accomplished by using an APL function named OUTLY (Appendix B). In order to minimize the impact of implementing our outlier identification and replacement policy on the integrity of the source data base, not all identified outliers are replaced using the described method. Many outliers are on the high side of the inventory distribution, and are assumed to be accurate values. Some outliers are on the low side but represent trends in the data or may only occur in a small number of MOS groups. We only adjust those outliers in which undercounts are suspected across the spectrum of all MOS groups according to the procedures described. Specifically, the cells determined to be faulty and selected for adjustment are shown in Table 1.

TABLE 1. OUTLIER IDENTIFICATION

PAYGRADE	OUTLIER PERIODS
2nd Lt	45
1st Lt	45
Captain	12 14 45
Major	8 12 14 45
LtCol	14 45

III. AGGREGATION AND SHRINKAGE PROCEDURES

A. GENERAL

This section summarizes the combined works of Misiewicz and Larsen as it pertains to cell aggregation and expansion procedures. Cell aggregation is the collection of cells possessing homogeneity of behavior with respect to attrition. In the original MCORP model, cells were aggregated by pooling several into a single cell in order to meet a user-defined minimum inventory threshold. This single cell was then used to determine the attrition rate estimate for the original, user targeted cell. Rather than aggregating into a single cell, the Larsen-Misiewicz' procedure pools cells into subsets of cells meeting user-defined specifications described below. This is necessary to provide the proper setting for the application of shrinkage techniques. Cell expansion provides the means by which cells are selected for aggregation so as to meet user specifications and "shrink" individual cell averages towards a more statistically stable "grand mean".

B. EXPANSION

Larsen's application of a hierarchical clustering algorithm to the NPRDC data set provided a major breakthrough in cell aggregation. His analysis developed the general idea of a hierarchy of MOS groups, with each Marine Corps primary

MOS belonging to one of fourteen small MOS groups, one of six large MOS groups, and one of four major MOS groups. Misiewicz' modifications to the original grouping are minimal and is displayed in Table 1 of Appendix C. Table 2 of Appendix C displays the YCS expansion bounds which reflect the maximum expansion allowed from the i. itial YCS defined cell.

The expansion process is an extension of the Small Cell Override Methodology used by NPRDC. Expansion involves finding more cells to be used to produce a number of cells with similar attrition characteristics. The end objective is to produce a collection of cells having moderate personnel inventories whose attrition rates can be "shrunk" towards the weighted grand mean. Greater stability for the attrition rates is achieved in this way. Expansion can be achieved using MOS and YCS. MOS expansion takes place on the range from the small group to the large group and then to the major MOS group. With YCS, we expand one year at a time over the allowable bound, usually in an alternating fashion. If the user-defined year is given as YCS_i, the expansion proceeds to YCS_{i-1}, YCS_{i+1}, YCS_{i-2}, YCS_{i+2}, etc. As the expansion process continues on the YCS scale, aggregation is recomputed at each step in the process.

The manpower planner initiates the shrinkage process by defining a cell for which an attrition rate estimate is required. He then defines the minimum cell inventory

threshold, denoted as T_c, as well as a minimum number of cells in the aggregate subset which must exceed the cell inventory threshold, denoted as K_c.

Since expansion is only made upon MOS groups and YCS, all other defined categories remain fixed. Whereas Misiewicz defines these categories as paygrade, service component and commissioning source (and additionally discusses sex, education and others), because of our data base, we have modified the algorithm to consider only MOS, YCS, paygrade and service component. The steps of the expansion are summarized in the following six stages:

- Stage 1 - Locate the small MOS group which contains the user-defined MOS. The MOSs in this group specify the initial cells for the user-defined YCS, paygrade, and service component. These cells are aggregated to obtain cells with average inventory greater than or equal to T_c. If the number of aggregated cells exceeds K_c, stop, otherwise continue to stage 2.

- Stage 2 - Expand by incrementing YCS one year at a time according to the bounds in Table 2 of Appendix C for all MOSs in the small MOS group. After each increment, aggregate and check to see if T_c and K_c are achieved. If so, stop, otherwise continue to increment YCS. If the YCS bound is reached before meeting user specifications, retain the cells identified in stages 1 and 2, and continue to stage 3.

- Stage 3 - Expand to the large MOS group for the single defined YCS, aggregate to attain cells with average inventory of at least T_c, then check to see if K_c cells have been achieved. If so, stop, otherwise continue to stage 4.

- Stage 4 - Expand by incrementing YCS for the large MOS group as was done in stage 2. After each increment of YCS, perform aggregation to obtain cells of minimum size T_c, then check to see if K_c cells are obtained. If so,

stop, otherwise continue to increment YCS. If the bound is reached before meeting user specifications, retain the cells identified and continue with stage 5.

- Stage 5 - Expand to the major MOS group for the defined YCS. Aggregate and check as in stage 3. If the specifications are not met, go to stage 6.
- Stage 6 - Expand by incrementing YCS for the major MOS group as was done for the large group in stage 4. If the YCS bound is reached before obtaining enough aggregated cells, stop. No more expansion is allowed. The user thresholds are unattainable.

It is important to note that the cells identified by previous stages are retained to subsequent stages to maintain the greatest degree of homogeneity. We desire to locate as many cells from the small MOS group as possible prior to expanding to the large MOS group. But, when aggregating cells, it is important to note that all prior aggregation is discarded. The pooling of all currently identified cells enhances greater flexibility and optimality in the aggregation algorithm.

C. AGGREGATION

While the expansion stages seek to achieve the threshold levels specified by the user, those cells with inventory less than T_0 must be gathered into larger, aggregated cells whose combined inventory meets or exceeds T_0. To limit the expansion to as few additional MOSs and YCSs as possible, we desire to maximize the number of aggregated cells at any stage

of the expansion. Misiewicz successfully employs a heuristic "greedy" algorithm to approximate optimality in aggregation.

A summary of the heuristic algorithm is as follows:

- Given a set of cells S, partition them into two subsets; S_1, consisting of cells of inventory greater than T_0, and S_2, consisting of cells of inventory less than T_0. Those cells of S_1 are then moved to the set K, and counted against K_0.

- The remaining cells in S_2 are ordered by inventory size. Selecting the cell of greatest inventory, c_n, find the smallest cell remaining which when united with c_n, results in a cell inventory at least T_0. This combined cell is then moved to K, and the process continues.

- If no single cell when combined with the cell having c_n exceeds T_0, then combine the two cells of greatest inventory, then find the smallest cell remaining which when united with these two cells, results in a cell inventory at least T_0, and so forth.

- Continue this procedure until the sum of all cells remaining in S_2 do not add to T_0. They are then aggregated to the cells in K according to order, e.g., the largest remaining cell in S_2 is aggregated to the smallest aggregated cell in K, etc.

- When the number of aggregated cells in K does not meet the specifications of K_0, expansion is required, all identified cells are retained, but the aggregation of cells is discarded.

Misiewicz used ad hoc methods for selecting values for T_0 and K_0, and limited the normal range from five to thirty. He further constrains T_0 to be less than or equal to K_0. Appendix B contains our modified FORTRAN version of the Misiewicz algorithm (MC90). This algorithm is suited to our data base and allows quarterly estimation of attrition rates.

IV. SEASONALITY

A. GENERAL

While the results obtained by Misiewicz show promising stability of estimators for a single year lead time, it is felt that better estimators for shorter periods may be achievable through analysis of seasonal behavior. A main advantage in applying the Harrison-Stevens approach to short-term forecasting (Harrison-Stevens, 1967) or the Winters' method of exponential seasonal smoothing (Makridakis and Wheelwright, 1978) is the incorporation of a value to account for variability between seasons. These methods are explained in Chapter V.

Common examples of seasonality are recognized when discussing monthly rainfall over a ten year period, quarterly home sales over a six-year period, or daily amusement park attendance over the eighteen-week summer. A snapshot look at the Marine Corps officer attrition data for any given year will show differing levels between the four quarters of that year. In general, we find that attrition rates are consistently highest during the third quarter of the calendar year and lowest during the first quarter of the year. This phenomena is present for many reasons, but may be generalized by two factors:

- There exists a higher number of contractual expirations during the summer months due to the high rate of entries during the summer.
- Many officers choose to terminate service during the summer months for family convenience, (e.g., when the children are out of school for the summer break).

Since efforts are directed toward establishing forecasts by some period of time, it would be desirable to find a seasonal factor which improves the forecast for each quarter. Some applications of ORG require monthly estimation.

For seasonality to be applicable, it is desirable to show that there is dependency between forecasting factors and seasonality. With periodic homogeneity (stationarity) from one season to the next, our seasonal factors $\{s_t(k), k=1,2,3,4\}$ of the Harrison-Stevens approach, or the seasonal index $\{S(k), k=1,2,3,4\}$ of the Winters forecasting method, would be equal to one. The variable k, used in conjunction with the seasonal factors, represents the four quarters of the year. Failing stationarity, our approach to seasonality analysis has been wide in scope, including use of:

- A single set of four seasonality values held constant over all periods (one for each quarter of the calendar year).
- A simple set of seasonality values which are updated over each subsequent period.
- Seasonality values which are updated based upon retention rates rather than attrition rates.
- A complex set of seasonality values which are cell (PG, MOS, YCS, and SC) specific and updated quarterly.

- Seasonal values S_k scaled with two alternative constraints given as:

$$\frac{\sum_{k=1}^{4} S_k}{4} = 1 \quad , \tag{4.1}$$

$$\prod_{k=1}^{4} S_k = 1 \ . \tag{4.2}$$

In our analysis of seasonality, we have looked at all of the above, and in numerous combinations. The least complicated method would be to calculate four constant seasonal values for officer attrition over the life of the system where the average of these values is equal to one. While this method, discussed in the next section, fails due to instability in the season values, it can be used to initialize other systems. The values are then updated at each period in the process. This procedure is discussed next. The final method discussed mirrors that of the preceding section with the main difference being that the product of the seasonal values equals one rather than the average equalling one. For the mathematical computations of this method, we are required to base the seasonal values on the officer continuation rate rather than the complementary attrition rate.

B. SEASONAL VALUES CONSTANT OVER CELL TYPES

The simplest approach in dealing with seasonality is to estimate a set of seasonal values for each of the four periods. We define p_{ijk} as the attrition rate for each MOS group/paygrade combination (i,j) and season k. Let p_{ij} be the quarterly average attrition rate over the four seasons. It is most convenient to have seasonal constants $s_1, \ldots, s_4$ which do not depend on (i,j), such that $p_{ijk} = p_{ij}s_k$. This also includes the special case of stationarity when all $s_k = 1$. As a formal test of this hypothesis we have

$$\begin{aligned} H_o&: p_{ijk} = p_{ij}s_k \\ H_a&: p_{ijk} \neq p_{ij}s_k \end{aligned} \tag{4.3}$$

where H_o is the null hypothesis.

In testing the hypothesis of equation (4.3), we estimate p_{ijk} with

$$\hat{p}_{ijk} = \frac{y_{ijk}}{n_{ijk}} \tag{4.4}$$

where y_{ijk} is the attrition values and n_{ijk} is the inventory values. Also,

$$\hat{p}_{ij} = \frac{\sum_k y_{ijk}}{\sum_k n_{ijk}} \tag{4.5a}$$

and

$$\hat{s}_k = \frac{4 \frac{y_{..k}}{n_{..k}}}{\sum_k \frac{y_{..k}}{n_{..k}}} \tag{4.5b}$$

where a subscript dot indicates summation over all values of that index. The s_k estimates are indicated in Table 2.

The test statistic is then computed and compared to the χ^2_{df}, or the normal approximation for large degrees of freedom. The test statistic is

$$TS = \sum_i \sum_j \sum_k n_{ijk} \frac{(\hat{p}_{ijk} - \hat{p}_{ij}\hat{s}_k)^2}{\hat{p}_{ij}\hat{s}_k(1 - \hat{p}_{ij}\hat{s}_k)} \tag{4.6}$$

with the degrees of freedom, df = IJK - 3 = 277.

TABLE 2. CONSTANT SEASONAL VALUES

QUARTERLY	1st QTR	2nd QTR	3rd QTR	4th QTR
Loss $y_{..k}$	3497	5733	6198	4870
Inv'try $n_{..k}$	3,792,668	3,922,789	3,838,841	3,816,026
$y_{..k}$ / $n_{..k}$	.0009220	.0014615	.0016188	.0012762
Seas'ty $\hat{s}_k$	0.6987	1.1075	1.2267	0.9671

It was decided to try this method (equation 4.3) for fixed ranks, i.e., separate tests for individual fixed paygrade j. Figure 1 gives a sample output comparing the test statistic to the computed χ^2 values for individual MOS group/paygrade combinations. It is evident that a constant set of seasonal values supports the null hypothesis when specifically forecasting attrition rates for the ranks of Major and Lieutenant Colonel, but fails due to instability in the lower officer ranks.

```
      HYPOTHESIS TESTS FOR SPECIFIC PAYGRADES
 (.05 Level of Significance -- 39 degrees of freedom)

    2ND LIEUTENANTS
         THE TEST STATISTIC IS         283.03798
         CHI SQUARE CRITICAL VALUE      54.51346

    IST LIEUTENANTS
         THE TEST STATISTIC IS          55.93979
         CHI SQUARE CRITICAL VALUE      54.51346

    CAPTAINS
         THE TEST STATISTIC IS          61.40886
         CHI SQUARE CRITICAL VALUE      54.51346

    MAJORS
         THE TEST STATISTIC IS          33.91452
         CHI SQUARE CRITICAL VALUE      54.51346

    LT COLONELS
         THE TEST STATISTIC IS          24.34368
         CHI SQUARE CRITICAL VALUE      54.51346
```

Figure 1. Hypothesis Test on Paygrades

Testing this hypothesis on multiple ranks in combination fared no better, with Majors and Lieutenant Colonels in combination being the only case in which the null hypothesis could not be rejected. Example outputs are displayed in Figure 2. The overall test of equation (4.6) failed as well.

```
     HYPOTHESIS TESTS FOR PAYGRADE COMBINATIONS
(.05 Level of Significance -- 81 degrees of freedom)

     IST LIEUTENANTS AND CAPTAINS
         THE TEST STATISTIC IS          122.93209
         CHI SQUARE CRITICAL VALUE      102.93406

     IST LIEUTENANTS AND MAJORS
         THE TEST STATISTIC IS          148.78286
         CHI SQUARE CRITICAL VALUE      102.93406

     CAPTAINS AND MAJORS
         THE TEST STATISTIC IS          131.91607
         CHI SQUARE CRITICAL VALUE      102.93406

     MAJORS AND LT COLONELS
         THE TEST STATISTIC IS           64.60649
         CHI SQUARE CRITICAL VALUE      102.93406

     CAPTAINS AND LT COLONELS
         THE TEST STATISTIC IS          117.27334
         CHI SQUARE CRITICAL VALUE      102.93406
```

Figure 2. Hypothesis test for Grade Combinations

C. PERIODIC UPDATES TO SEASONAL VALUES WITH A MEAN OF ONE

In many applications, seasonal factors tend to be constant from year to year. When viewing the trend of demand for a product, or the trend of attrition rates from the Marine

Corps, changes of season patterns are likely but inherently difficult to detect quickly. When seasonal factors change, means of modifying them are required.

To initialize seasonal values, the maximum likelihood estimator discussed in the previous section is used. Following each periodic forecast, we update all four season values as follows. Let:

d_t = current attrition rate,
m_t = current trend value,
k = season {k = 1,2,3,4},
A_3 = seasonal update factor.

Let:

$$e_k = \frac{d_t}{m_t} - \hat{s}_k \tag{4.7}$$

represent the difference between the crude estimate d_t/m_t from the current data, and the current seasonal estimate $\hat{s}_k$. We update the four seasonal values as follows:

$$New \quad \hat{s}_z = \begin{cases} \hat{s}_z + A_3 e_k & (z = k), \\ \hat{s}_z - \dfrac{A_3 e_k}{3} & (z \neq k). \end{cases} \tag{4.8}$$

For quarterly data, Harrison and Scott find values of A_3 in the range 0.1 to 0.3 useful (Harrison and Scott,1965).

D. MULTIPLICATIVE SEASONALITY PROCESS ON CONTINUATION RATES

Since the modelling of the attrition process can be viewed as Bernoulli trials of a Binomial process, it is natural to consider a multiplicative version of the seasonality adjustment from year to year. Further studies by Harrison and Scott find that the multiplicative model may be more suited for most seasonal data. Because of the computations involved, we are required to view this process from a continuation rate rather than attrition rate perspective. The rationale is as follows:

Let q = 1 - p be the yearly continuation rate, where p is the yearly attrition rate; let n_k be the quarterly personnel inventory values; and let y_k be the personnel losses for the quarters where $\{k = 1,2,3,4\}$. The estimated quarterly continuation rates are computed as:

$$\hat{q}_k = 1 - \frac{y_k}{n_k} \qquad for \quad k=1,2,3,4 \tag{4.9}$$

and, by independence of time periods, the estimated yearly continuation rate is:

$$\hat{q} = \prod_{k=1}^{4} \hat{q}_k . \tag{4.10}$$

Any inhomogeneities in the quarterly rates are attributed to seasonal factors; therefore:

$$q_k = q^* s_k \quad \text{where} \quad q^* = \sqrt[4]{q} \tag{4.11}$$

is the seasonally adjusted quarterly continuation rate. It follows that:

$$q = \prod_{k=1}^{4} q_k = \prod_{k=1}^{4} q^* s_k = q \prod_{k=1}^{4} s_k \tag{4.12}$$

which implies that the product of the seasonal factors is equal to one.

A basic and initialization estimate of the $\{s_k\}$ values can be made using modified minimum χ^2 procedures. Specifically,

$$\chi^2 = \sum_k n_k \frac{(q_k - s_k q^*)^2}{p_k q_k}$$
$$\cong \sum_k n_k \left[\frac{q_k}{p_k} - \frac{s_k q^*}{p_k} \right]. \tag{4.13}$$

A LaGrangian term is included to treat the constraint. So minimize

$$\Phi = \sum_k n_k \left[\frac{q_k}{p_k} - \frac{s_k q^*}{p_k} \right] + \lambda \sum_k \ln(s_k) , \tag{4.14}$$

then, the estimators which minimize (4.14) satisfy

$$\hat{\lambda} = \frac{\hat{s}_k n_k \hat{q}^*}{\hat{p}_k}, \qquad k=1,2,3,4 . \tag{4.15}$$

Since the product of the four seasonal values is equal to one, equation (4.15) can be multiplied over all seasons to obtain

$$\hat{\lambda}^4 = \prod_k \hat{s}_k \prod_k \left(\frac{n_k \hat{q}^*}{\hat{p}_k} \right) = \hat{q} \prod_k \frac{n_k}{\hat{p}_k} . \tag{4.16}$$

From equations (4.15) and (4.16), we can solve:

$$\hat{s}_k = \frac{\lambda \hat{p}_k}{n_k q^*} ,$$

$$\hat{\lambda} = \sqrt[4]{\prod_k \frac{(n_k - y_k)}{\hat{p}_k}} , \qquad (4.17)$$

$$\hat{q}^* = \sqrt[4]{\prod_k \frac{(n_k - y_k)}{n_k}} .$$

Through reduction, the seasonal values may bc computed using equation (4.17) to form equation (4.18):

$$\hat{s}_k = \frac{y_k}{n_k^2} \sqrt[4]{\prod_k \frac{n_k^2}{y_k}} , \qquad (4.18)$$

or we may accept the ad hoc estimation based upon equation (4.19) (which we choose to do in our model) to solve the seasonal values in equation (4.20):

$$\hat{s}_k \hat{q}^* = \hat{q}_k = \frac{n_k - y_k}{n_k} , \qquad (4.19)$$

$$\hat{s}_k^* = \frac{\hat{q}_k}{\hat{q}^*} = \frac{\dfrac{n_k - y_k}{n_k}}{\sqrt[4]{\prod_k \dfrac{n_k - y_k}{n_k}}} \; . \tag{4.20}$$

The multiplicative version of periodic updating is weighted with its prior calculation and is generally:

$$s_t = (s_t^*)^{\beta} (s_{t-4})^{1-\beta} \tag{4.21}$$

for some $0 < \beta < 1.0$, where s_{t-4} is the seasonal value for the previous year during this quarter, and S_t^* is the seasonal adjustment based upon the immediate data.

As this method is based upon continuation rates, the attrition rate forecast is simply found from:

$$p_k = 1 - q^* s_k \; . \tag{4.22}$$

Some final notes need to be made regarding the mathematical feasibility of this approach. Obviously, there must be a positive inventory value for each n_k or else we would be attempting division by zero when computing the continuation rates. Additionally, the continuation rate cannot be zero (attrition of the entire cell inventory), as it is used in the denominator when updating the seasonal values.

We observe six instances out of 3080 MOS/paygrade/season data observations where the continuation rate equals zero ($y = n = 1$). We choose to compensate in these irregular instances using the LaPlace Law of Succession whereby the inventory is incremented by two and the continuation is incremented by one, so that we have

$$\hat{q}_k = 1 - \hat{p}_k = \frac{n_k - y_k + 1}{n_k + 2} \quad . \tag{4.23}$$

V. ESTIMATION METHODS

A. WINTERS THREE PARAMETER SEASONAL EXPONENTIAL SMOOTHING

Exponential smoothing methods are appropriate for time series that have a constant mean or a mean that changes gradually with time. Three linear exponential methods are examined by Makridakis and Wheelwright in an attempt to deal directly with non-stationary time series that exhibit a significant trend. They differ from single exponential smoothing in that they introduce additional formulas that estimate the trend so that it can be subsequently used to improve forecasting efforts.

In developing the Winters model we build upon Brown's One Parameter Linear Exponential Smoothing which was used by Hogan. With D_t given as the attrition rate in period t, the equations used in Brown's model are:

$$E'_t = \alpha D_t + (1 - \alpha) E'_{t-1} \qquad (5.1)$$

$$E''_t = \alpha E'_t + (1 - \alpha) E''_{t-1} \qquad (5.2)$$

where E_t' is the single exponentially smoothed value of D in time t, and E_t'' is the double exponentially smoothed value of D for that time period;

$$b_t = \frac{\alpha}{1-\alpha}\,(E'_t - E''_t) \tag{5.3}$$

where b_t is an estimate of trend;

$$a_t = E'_t + (E'_t - E''_t) = 2E'_t - E''_t \tag{5.4}$$

where a_t is an estimate of the intercept; and finally, the forecasts are found using:

$$D^*_{t+u} = a_t + ub_t \tag{5.5}$$

where u is the number of periods ahead to be forecast.

The first equation is simply the formula used for single exponential smoothing. The next serves to smooth the values of the first equation. It is introduced to estimate the trend through the concept of lagging values. Equation (5.3) divides the factor α by $1-\alpha$, then multiplies by the difference between the single and double exponential smoothing values. This results in a trend for a single period. Equation (5.4) then makes an estimate for the present level intercept of the data using the same concept of equation (5.3). In order to forecast, equation (5.5) is used starting from the current level, a, and adding as many times the trend, b, as the number of periods ahead one wants to forecast. This is, therefore,

a direct adjustment for the trend factor which may exist in the data.

As with Brown's method, Holt's two parameter linear exponential smoothing method estimates and uses the trend in forecasting. The difference in these two methods is that Holt introduces a term for the trend (T_{t-1}) and an additional smoothing constant β. The three equations in Holt's model are:

$$E_t = \alpha D_t + (1 - \alpha)(E_{t-1} + T_{t-1}), \tag{5.6}$$

$$T_t = \beta(E_t - E_{t-1}) + (1 - \beta)T_{t-1}, \tag{5.7}$$

$$D^*_{t+u} = E_t + uT_t. \tag{5.8}$$

Holt uses the difference between two successive exponential smoothing values, which have been smoothed for randomness in equation (5.7), to estimate the trend in the data. Using the smoothing constant, β, multiplied by this difference, and $(1-\beta)$ by the old estimate, we get the smoothing trend which includes reduced randomness. To compute the forecast in equation (5.8), the trend is then multiplied by the number of periods ahead that one desires to forecast and then the product is added to E_t which is the current level of the data that has been smoothed to eliminate randomness.

In comparison with Brown's model, Holt's model has the disadvantage of requiring two parameter specifications (α and β) whose values need be optimized if the mean squared error is to be minimized. On the other hand, one has the opportunity of applying different weights to randomness and trend depending upon the specific data involved.

Winters' exponential smoothing is an extension of Holt's linear exponential smoothing. Its applicability in our study is its inclusion of a seasonality factor.

The estimate of seasonality is given by an index S_t, which fluctuates around the value of 1. The equations in Winters' method are:

$$S_t = \beta \frac{D_t}{E_t} + (1 - \beta) S_{t-L} , \tag{5.9}$$

$$E_t = \alpha \frac{D_t}{S_{t-L}} + (1 - \alpha) (E_{t-1} + T_{t-1}) , \tag{5.10}$$

$$T_t = \gamma (E_t - E_{t-1}) + (1 - \gamma) T_{t-1} , \tag{5.11}$$

$$D^*{}_{t+u} = (E_t + uT_t) S_{t-L+u} . \tag{5.12}$$

The form of equation (5.9) is similar to that of other exponential smoothing equations, i.e., a value is multiplied

by a smoothing constant β, and is then added to its previous estimate multiplied by $(1 - \beta)$. D_t/E_t is used rather than either variable independently so as to express the value as an index rather than in absolute terms. Winters' equations differ from Holt's in the introduction of the seasonal index S_t. Thus, equations (5.10), (5.11) and (5.12) obtain estimates of the present level of the data, the trend, and the forecast for some future period $(t + u)$. To remove the seasonal effects which may exist in the original data D_t, equation (5.10) has D_t divided by the seasonal index S_{t-L}, where L is the length of seasonality, or number of periods experienced before returning to a period with similar characteristics. A forecast is then obtained in equation (5.12) in a similar manner to that used by Holt. However, this estimate for the future period $(t + u)$, is multiplied by the last seasonal index S_{t-L+u}, to readjust the forecast for seasonality. Our Winters FORTRAN algorithm is included in Appendix B.

The Winters' model is more difficult to optimize because it has three parameters. Values for the randomness smoothing constant α, the seasonality smoothing constant β, and the trend smoothing constant γ must be found to minimize the mean squared error. Makridakis and Wheelwright suggest that the values for β and γ are usually smaller than α. They suggest a normal α value ranging from 0.1 to 0.3. Hogan correctly

recognized instability of an optimal α when viewing the spectrum of MOS groups. For a single value α, he reluctantly suggests 0.4 be used. Through our own sensitivity analysis discussed in the next chapter, we chose to select values for α of 0.45, β of 0.35, and γ of 0.10 though we admit that there is room for future analysis and refinement of these estimations.

B. HARRISON-STEVENS MULTI-PARAMETER ESTIMATION

P. J. Harrison and C. F. Stevens of Imperial Chemical Industries, Ltd., describe a method of short-term forecasting based on the use of Bayesian principles in conjunction with a multi-state data-generating process (Harrison and Stevens, 1971). The various states correspond to the occurrence of transient errors and step changes in trend and slope.

1. Basic Model

For the basic model, we define:

d_t = posterior attrition rate,
m_t = posterior trend value,
b_t = posterior slope value,
s_t = posterior seasonal factor.

Then the basic model is a generating process defined by:

$$\begin{aligned} d_t &= m_t s_t + \epsilon_t & (\epsilon \sim N(0\,;\, V_\epsilon)) \\ m_t &= m_{t-1} + b_t + \gamma_t & (\gamma \sim N(0\,;\, V_\gamma)) \\ b_t &= b_{t-1} + \delta_t & (\delta \sim N(0\,;\, V_\delta)) \end{aligned} \tag{5.13}$$

where:

ϵ_t = observational noise,
γ_t = trend perturbation,
δ_t = slope perturbation,

and the random components ϵ, γ, and δ are assumed to be independently and normally distributed with zero means and known, but not necessarily constant, variances V_ϵ, V_γ, and V_δ. The posterior distribution is used to assess errors in the forecasts. Also, as is usual in Bayesian procedures, the posterior values become the prior values for the time step update.

Harrison has shown that given a generating process of this type with constant variances, and ignoring the seasonal effect s_t, the optimal least-squares predictor is equivalent to that of the Holt system described earlier (Harrison, 1967). In the Harrison-Stevens model, we change notation slightly, replacing α and β with the smoothing constants A_1 and A_2, these being functions of the variance ratios; V_γ/V_ϵ and V_δ/V_ϵ.

Normally the parameters A_1 and A_2 determine the sensitivity of a system. Conflict arises between a sensitive system which responds quickly to real changes, and an insensitive system which does not react to noise and transient errors. One is more likely to overswing while the other is too slow to catch up with the data. Since we are more likely to experience transient error than changes in trend or slope, we err on the side of insensitivity. When large changes occur, we either accept slow correction to the desired level

or utilize a method of monitoring forecast errors with manual adaptivity. The Bayesian multi-state system is capable of overcoming these difficulties, being adaptive to trend and slope as well as responsive to transients.

2. The Multi-State Model

In equation (5.13), we see that the generating process is characterized by the noise component ϵ_t which affects only the current observation. A large ϵ_t has the appearance of a large transient error with no effect on the future of the system. We also have the perturbation terms γ_t and δ_t which affect the future course of the system. A large γ_t causes a permanent step change to a new level, and a large δ_t causes a change in slope. The multi-state model supposes that there is not one but a number of possible distributions from which these values are generated at each observation. Since we have distinguished four process states; no change, step change, slope change, and transient, we formalize the multi-state model as follows:

π_j = probability of occurrence of the jth state $\{j = 1, 2, 3, 4\}$

$$\begin{aligned} \epsilon_t &\sim N(0\,;\,V_\epsilon^{(j)}) \\ \gamma_t &\sim N(0\,;\,V_\gamma^{(j)}) \\ \delta_t &\sim N(0\,;\,V_\delta^{(j)}) \end{aligned} \qquad \textbf{(5.14)}$$

which produces the random components ϵ_t, γ_t, and δ_t when the system is in state j at time t. As recommended by Harrison and Stevens, we define the variances in terms of ratios of the basic noise V_0, a value of the basic variability of the process in its normal state. Testing the sensitivity of this variance law, Harrison-Stevens applied a range of incorrect values and experienced minimal forecasting impairment when only a short stabilizing lead interval was provided (one to five forecast periods). As amplified in Chapter VI, we could not verify these findings and instead rely on the variance law obtained from Misiewicz. The variances are then defined as follows:

$$\begin{aligned} V_e^{(j)} &= R_e^{(j)}\, V_0 \\ V_\gamma^{(j)} &= R_\gamma^{(j)}\, V_0 \\ V_\delta^{(j)} &= R_\delta^{(j)}\, V_0 \end{aligned} \tag{5.15}$$

using the parameters defined in Table 3.

Given this type of generating system we can never know the values of m_t or b_t at any time t. But we can express our knowledge about m_t and b_t in terms of a distribution which is continually modified with each successive attrition rate observation d_t, d_{t+1}, . . .

TABLE 3. STATE PARAMETERS IN THE HARRISON-STEVENS MODEL

State	Prob.	R_ϵ	R_γ	R_δ
No Change	0.900	1	0	0
Step Change	0.003	1	100	0
Slope Change	0.003	1	0	1
Transient	0.094	101	0	0

With the generating process of equation (5.13), the joint distribution of (m,b) at time (t - 1) is bivariate normal, as is the posterior distribution at time t.

In developing the joint distribution, we let:

$$e_t = d_t - (m_{t-1} + b_{t-1})\, s_t, \tag{5.16}$$

$$R = \begin{bmatrix} r_{11} & r_{12} \\ r_{12} & r_{22} \end{bmatrix}, \tag{5.17}$$

and:

$$V = \begin{bmatrix} V_{mm} & V_{mb} \\ V_{mb} & V_{bb} \end{bmatrix}, \tag{5.18}$$

where:

$$\begin{aligned} r_{11} &= V_{mm} + 2\,V_{mb} + V_{bb} + V_\gamma + V_\delta \\ r_{12} &= V_{mb} + V_{bb} + V_\delta \\ r_{22} &= V_{bb} + V_\delta\,. \end{aligned} \tag{5.19}$$

Further:

$$
\begin{aligned}
V_e &= S_t^2 r_{11} + V_e , \\
A_1 &= \frac{S_t r_{11}}{V_e} , \\
A_2 &= \frac{S_t r_{12}}{V_e} .
\end{aligned}
\tag{5.20}
$$

We then have the joint posterior distribution at time t given by:

$$
\begin{aligned}
m_t &= m_{t-1} + b_{t-1} + A_1 e_t \\
b_t &= b_{t-1} + A_2 e_t \\
V_{mm,t} &= r_{11} - A_1^2 V_e \\
V_{mb,t} &= r_{12} - A_1 A_2 V_e \\
V_{bb,t} &= r_{22} - A_2^2 V_e .
\end{aligned}
\tag{5.21}
$$

At this point, we introduce the distribution notation use by Harrison-Stevens to formalize the relationship between the prior and posterior distributions. In the multi-state model where we have a mixed prior distribution specified by:

$$
(m_{t-1}, b_{t-1}) \sim \sum_{j=1}^{n} q_{t-1}^{(j)} N(\phi_{t-1}^{(j)}) \ , \tag{5.22}
$$

one component of the prior corresponds to each state of the process, with:

$q_{t-1}^{(j)}$ = probability (posterior to d_{t-1}) that the process was in state j at time step (t-1),

$\phi_{t-1}^{(j)}$ = state j parameter values at time step (t-1).

We then complete the posterior distribution as:

$$(m_t, b_t | d_t) \sim \sum_{i,j} q_t^{(i,j)} \, N(\phi_t^{(i,j)}) \ , \tag{5.23}$$

where $\phi_t^{(i,j)}$ is obtained from $\phi_{t-1}^{(i)}$, $V_\epsilon^{(j)}$, $V_\gamma^{(j)}$, and $V_\delta^{(j)}$, and $p_t^{(i,j)}$ is the state transitional matrix developed by:

$$p = q_{t-1}^{(i)} \pi_j \left\{ \exp \frac{-e_j^2}{2 V_e} \right\} \frac{1}{2\pi V_e} \ . \tag{5.24}$$

A complex mathematical problem develops when an 'N-component' prior proceeds through this process to become an 'N^2' posterior. As we continue to generate further, it becomes N^3, N^4, etc. To overcome this mathematically correct yet complex process, the posteriors are condensed to create an approximate bivariate normal distribution of the same form as the prior distribution. Using subscripts, we show the condensed state probability and bivariate values of the trend and slope:

$$
\begin{aligned}
q_t^{(j)} &= \sum_i p_t^{(i,j)} \,, \\
m_t^{(j)} &= \frac{\sum_i p_t^{(i,j)} m_t^{(i,j)}}{q_t^{(j)}} \,, \\
b_t^{(j)} &= \frac{\sum_i p_t^{(i,j)} b_t^{(i,j)}}{q_t^{(j)}} \,,
\end{aligned}
\tag{5.25}
$$

and one example from the variance-covariance matrix calculated from the multivariate values:

$$
v_{mm,t}^{(i)} = \frac{\sum_i p_t^{(i,j)} \; [v_{mm,t}^{(i,j)} + (m_t^{(i,j)} - m_t^{(j)})^2]}{q_t^{(j)}} \,. \tag{5.26}
$$

With this process the more relevant information corresponding to the current process state is carried forward, and the posterior is in the proper form for the process to continue indefinitely.

Finally, our forecast for time t is calculated by:

$$
d_t^{*} = \sum_j q_t^{(j)} \; [m_t^{(j)} + b_t^{(j)}] \; s_t \,. \tag{5.27}
$$

Our FORTRAN algorithm is contained in Appendix B.

VI. VALIDATION PHASE

A. GENERAL

With two alternative estimation techniques, we require a method of determining their individual effectiveness as well as their relative effectiveness to the forecasting model. We employ two Measures of Effectiveness (MOE) to achieve these results:

- Mean Squared Error (MSE)--an average measure of the difference between the forecast attrition and the actual attrition rate after being squared.
- Mean Absolute Deviation (MAD)--an average measure of the magnitude difference between the forecast attrition and the actual attrition rate in absolute terms.

Each MOE is dependant upon the difference between the actual period attrition rate and the model forecast attrition rate. To validate the presented estimation techniques, we weighted the differences between the actual attrition rates and the forecast attrition rates. Assuming the forecast has negligible variance, then:

$$Var(A-F) \approx Var(A) = \frac{pq}{n},$$

where:

A = Actual attrition rate,
F = Forecast attrition rate,
n = Inventory level of the forecasting cell,

n = Inventory level of the forecasting cell,
p = Probability of attrition, and
q = 1 - p.

Then the normalized Error of Forecast (EOF) is

$$EOF = (A - F) \times \sqrt{\frac{n}{p \times q}} . \qquad (6.1)$$

Values for p and q are lacking, but the product 'pq' should not vary much; certainly not as much as n. We therefore modify our calculation for EOF as:

$$EOF = (A - F) \times \sqrt{n} \qquad (6.2)$$

which provides a more stabilizing verification value than the simple difference. This EOF is the foundation of remaining validation.

The validation phase encompasses the following objectives:

- Compute values for the constant parameters presented in both the Harrison-Stevens and the Winters techniques.

- Compare Harrison-Stevens and Winters forecast results. Design an experiment to compare the performance between the following four treatments: The Winters Exponential Smoothing technique with seasonality updates; Winters technique without using a seasonal factor; Harrison-Stevens Multi-parameter Estimation technique with seasonal updates; and Harrison-Stevens technique without using seasonal factors.

- At each observation period, compute a forecast for the subsequent four periods. Analyze the forecast distributions when estimating further into the future.

Since the data available is used both to establish parameter criteria and to test the model techniques, a means of cross-validation is required. Additionally, assuming possible error in setting initial parameters, some lead time is required to allow the process to stabilize. Since we are working with 48 periods, our ad hoc solution is to use the first eight periods to stabilize the process. The next ten periods are used to select and tune parameters, and the final 30 periods are used for cross-validation of our forecasting results.

B. PARAMETER ESTIMATION

1. Winters Parameters

The three constants included in the Winters Exponential Smoothing Technique are: α, the randomness smoothing constant; β, the seasonal smoothing constant; and γ, the trend smoothing constant. Our objective is to select those constant values which tend to minimizing the difference between the actual and forecast attrition rates over the wide range of paygrade/MOS group combinations. Lacking time to explore the possibilities of an optimization algorithm and knowing from Hogan's experience that it is unlikely that an optimal solution would approach a single set of values, we explore forecasting results through nested DO LOOPS within our forecasting algorithm. Figures 3 shows a sample output of the minimum MAD obtained for a given set of constants for a

particular MOS group and paygrade (MOS group 3; rank of Major). The low values tend to cluster around the minimum value. This is consistent for all MOS/paygrade combinations tested, but unfortunately, not all MOS/paygrade outputs cluster about the same constant values. It is difficult to select one set of values, but as in Hogan's experience, we reluctantly do so, and set $\alpha = 0.45$, $\beta = 0.35$, and $\gamma = 0.10$ for our cross validation.

2. Harrison-Stevens Parameters

Harrison and Stevens present a number of parameters, most of which we accept as given. Since we use the multiplicative seasonality method, the two values of most concern are V_0, the basic variance law for attrition, and β, the seasonality update weighting value.

As previously mentioned, we are unable to confirm the Harrison-Stevens claim that a minor lead time compensates for V_0 selection error. From Misiewicz, we estimate the true value of V_0 to be approximately 0.01. Through sensitivity analysis of the first few small MOS groups, we observe MAD value fluctuations from 3% to 14% for varying values of V_0. With all other values held constant, results are compared for V_0 set equal to 0.0001, 0.001, 0.01, and 0.1. Complicating matters still, V_0 and β proved strongly correlated when repeating the clustering parameter optimization procedures described above for the Winters parameters. We choose to rely

on the efforts of Misiewicz to support our single value estimate for $V_0 = 0.008$. With this approximation, the results prove most stable with our reluctant but quite believable single value estimate for $\beta = 0.40$.

WINTERS VALIDATION OF CONSTANTS USING MAD
(EXAMPLE USING MAJORS IN MOS GROUP 3)

FOR ALPHA = 0.2

GAMMA VALUES

BETA	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
0.1	0.200	**0.161**	**0.141**	0.213	0.202	0.327	0.230	0.263
0.2	0.192	**[illegible]**	**0.143**	0.355	0.162	0.479	0.206	0.292
0.3	0.188	**0.160**	**0.155**	0.483	0.234	0.555	0.330	0.641
0.4	0.187	**0.165**	0.191	0.884	0.336	0.214	0.237	0.347
0.5	0.196	0.196	0.245	1.959	0.286	0.306	0.311	0.271
0.6	0.227	0.248	0.330	1.184	0.310	0.345	0.310	0.902

FOR ALPHA = 0.3

GAMMA VALUES

BETA	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
0.1	**0.163**	**0.141**	**0.137**	**0.138**	**0.157**	**0.179**	0.344	0.308
0.2	**0.163**	**0.144**	**0.142**	**0.176**	0.250	1.076	0.374	0.288
0.3	**0.165**	**0.150**	**0.176**	0.244	0.847	0.296	0.245	0.712
0.4	**0.173**	0.182	0.228	0.394	0.310	0.335	0.665	0.257
0.5	0.203	0.229	0.312	0.908	0.255	0.376	0.271	0.315
0.6	0.255	0.309	0.467	0.554	0.361	0.338	0.560	1.551

FOR ALPHA = 0.4

GAMMA VALUES

BETA	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
0.1	**0.150**	**0.138**	**0.137**	**0.136**	**0.137**	**0.142**	**0.149**	0.181
0.2	**0.154**	**0.141**	**0.142**	**0.150**	**[illegible]**	**0.173**	0.185	0.191
0.3	**0.161**	**0.155**	**0.170**	0.157	[illegible]	0.221	0.272	0.362
0.4	0.180	0.190	0.217	0.250	[illegible]	0.306	0.455	0.506
0.5	0.220	0.245	0.293	0.353	0.416	0.482	0.769	0.296
0.6	0.268	0.339	0.426	0.555	0.717	0.831	0.573	0.873

FOR ALPHA = 0.5

GAMMA VALUES

BETA	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
0.1	**0.148**	**0.138**	**0.137**	**0.138**	**0.141**	**0.148**	0.174	0.178
0.2	**0.154**	**0.145**	**0.146**	**0.153**	**0.162**	0.176	0.990	0.180
0.3	**0.167**	**0.162**	0.172	0.184	0.197	0.235	0.212	0.313
0.4	0.192	0.197	0.210	0.227	0.248	0.366	0.264	0.257
0.5	0.235	0.252	0.275	0.298	0.333	0.308	0.403	0.380
0.6	0.311	0.346	0.387	0.428	0.506	0.600	0.575	0.583

FOR ALPHA = 0.6

GAMMA VALUES

BETA	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
0.1	**0.149**	**0.141**	**0.143**	**0.148**	**0.156**	**0.168**	0.209	0.185
0.2	**0.158**	**0.153**	**0.158**	**0.166**	0.178	0.225	0.200	0.255
0.3	0.178	0.176	0.193	0.195	0.223	0.430	0.245	0.226
0.4	0.205	0.208	0.220	0.240	0.290	0.247	0.324	0.331
0.5	0.250	0.261	0.279	0.310	0.423	0.327	0.353	0.394
0.6	0.328	0.352	0.380	0.433	0.966	0.647	0.723	0.569

ALPHA = 0.4 BETA = 0.1 GAMMA = 0.4
THE MINIMUM MAD IS: 0.136

Figure 3. Winters Minimum MAD Clusters

C. TECHNIQUE COMPARISON

In order to obtain some quantitative worth of seasonality, we compare each estimation technique while using seasonality factors against the same techniques without the use of seasonality. Figures 4 and 5 display the measure of effectiveness for 70 cases (14 MOS groups times five paygrades). As expected, seasonality improves the forecast for most MOS/paygrade combinations as measured by the MAD and MSE. However, the degree of improvement is not as great as that which is expected, and there exist cases where nonseasonality outperforms seasonality techniques. Figure 6 displays the comparison when seasonality factors are used between the Winters and the Harrison-Stevens techniques. Again, the resulting differences are not as great as expected, with Harrison-Stevens holding a slight edge.

To measure whether this difference is significant at the 90% confidence level, an Analysis of Variance is performed, with the results displayed in Figure 7. With the knowledge obtained from the MOE plots, it is not surprising that ANOVA concludes that we cannot reject the null hypothesis, i.e., the techniques cannot be separated statistically.

D. FORECASTS BEYOND THE NEXT PERIOD

The final area of analysis examines forecasts beyond the next calendar period. Without a seasonality factor, future

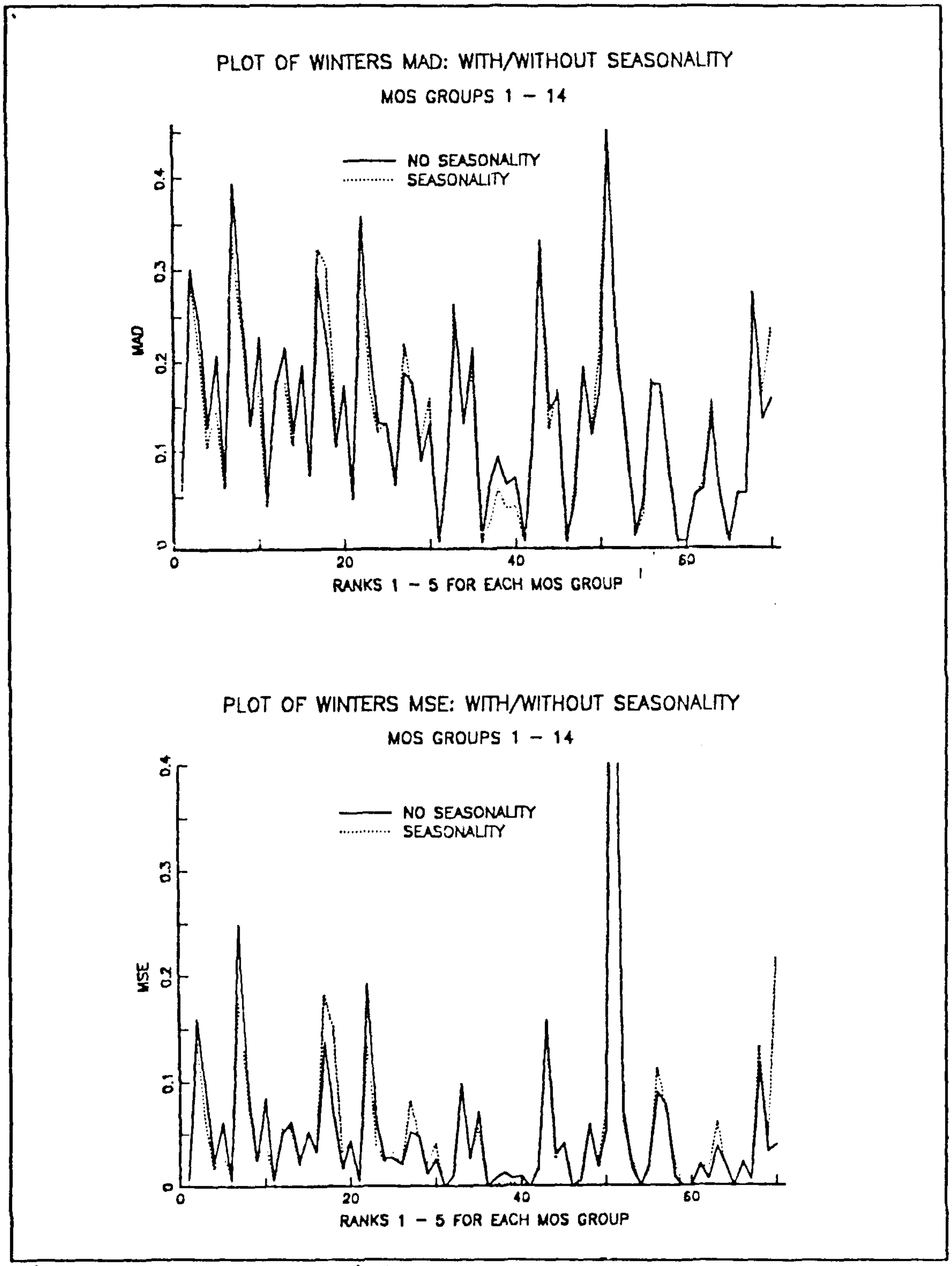

Figure 4. Plots of Winters MOEs

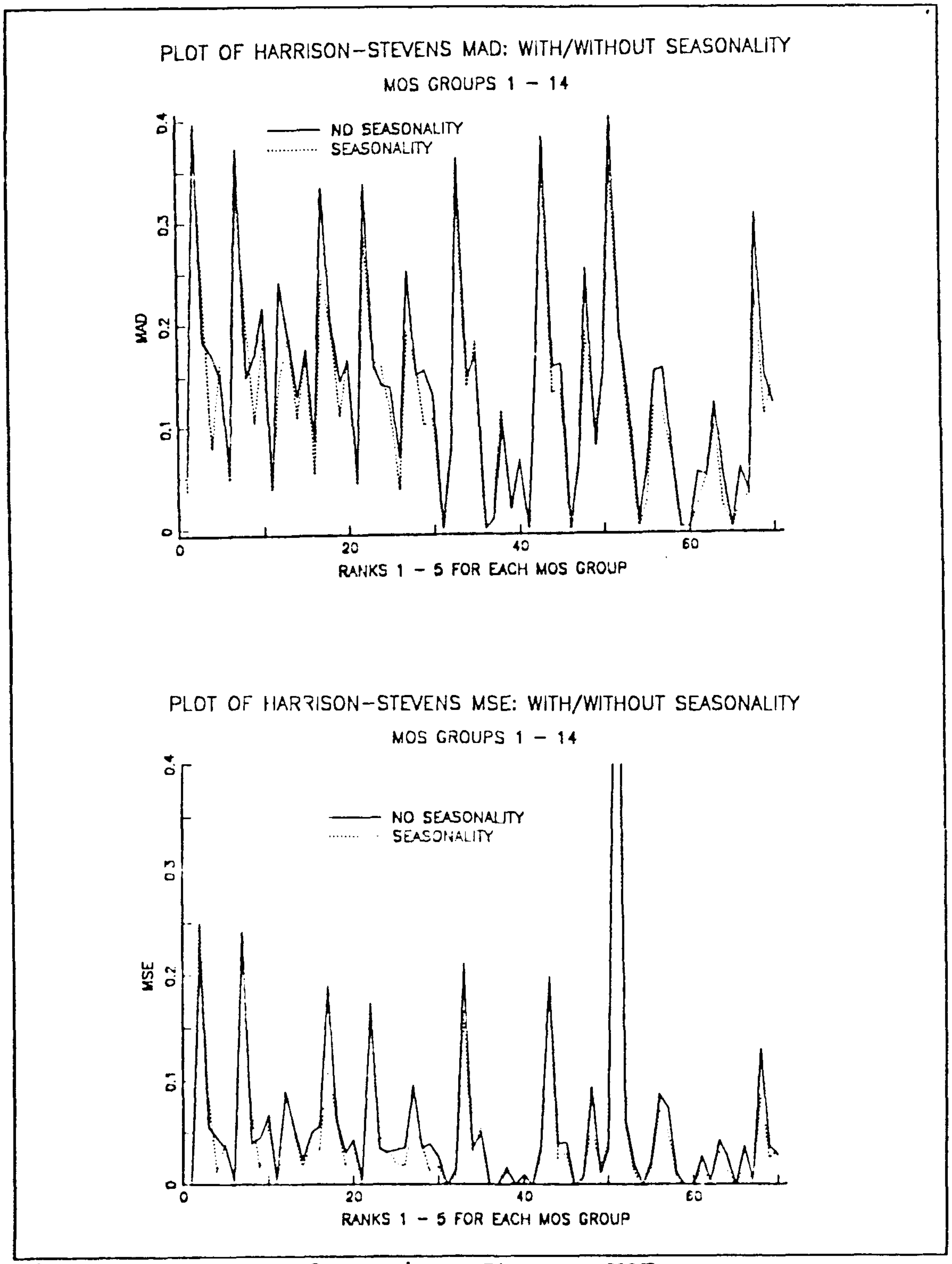

Figure 5. Plots of Harrison-Stevens MOEs

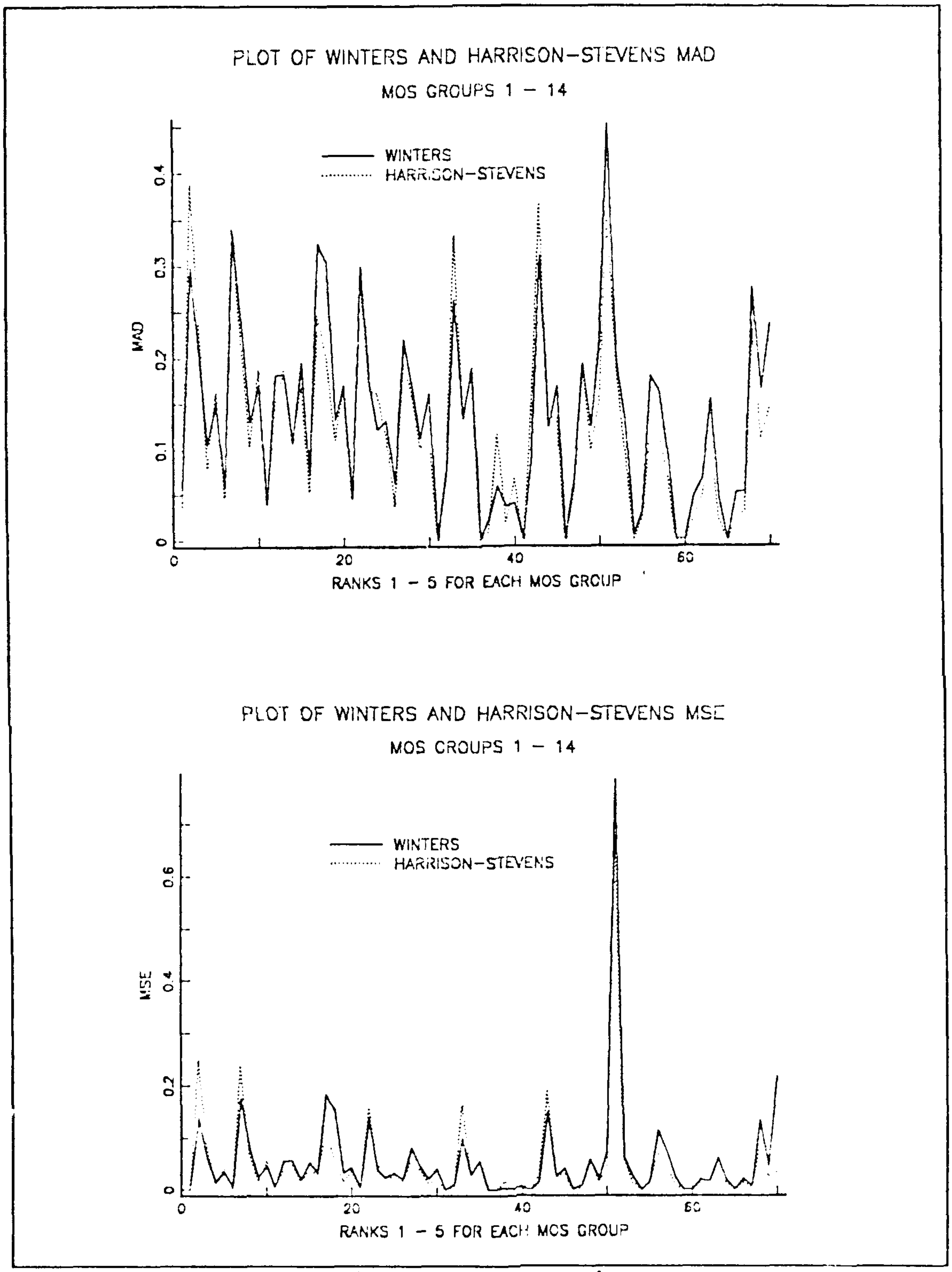

Figure 6. Plots of Winters vs. Harrison Stevens MOEs

ANOVA TABLE: BETWEEN ALL FOUR TREATMENTS

SOURCE	SUM OF SQR	DF	MEAN SQR	F
BETWEEN	0.002965157	3	0.0009883857	0.10402369
WITHIN	2.622426057	276	0.0095015437	
TOTAL	2.625391214	279		

ANOVA TABLE: BETWEEN WINTERS TREATMENTS

SOURCE	SUM OF SQR	DF	MEAN SQR	F
BETWEEN	0.000531796	1	0.000531797	0.055200235
WITHIN	1.329485192	138	0.009633951	
TOTAL	1.330016988	139		

ANOVA TABLE: BETWEEN HARRISON-STEVENS TREATMENTS

SOURCE	SUM OF SQR	DF	MEAN SQR	F
BETWEEN	0.001895421	1	0.001895421	0.202304747
WITHIN	1.292940865	138	0.009369137	
TOTAL	1.294836286	139		

ANOVA TABLE: BETWEEN THE TWO SEASONALITY TREATMENTS

SOURCE	SUM OF SQR	DF	MEAN SQR	F
BETWEEN	0.002469978	1	0.0024699780	0.257012677
WITHIN	1.326226277	138	0.0096103353	
TOTAL	1.328696255	139		

Figure 7. ANOVA Between Differing Forecast Techniques

projections are linear with trend-dependant slope. Multiplicative seasonality allows us to shed linearity in favor of a weighted forecast which is dependant upon the seasonal factor.

Winters and Harrison-Stevens are each capable of forecasting as far into the future as the user desires. It is natural to assume that the further into the future one forecasts, the less reliable the results become. We test the Winters and Harrison-Stevens models using 30 periods of data. For each period t, a forecast is made for the following four periods, t+1, t+2, t+3, and t+4. Figures 8 displays a partial output of the Harrison-Stevens EOFs obtained for a particular

MOS group/paygrade combination. The following example provides a visualization of improving forecast in the nearer periods. Compare the EOFs when forecasting for the 11th period. In period t = 7, the t+4 EOF represents the forecast error for period 11. In period t = 8, the t+3 EOF now represents period 11. In period t = 9, the EOF is t+2, and in period t = 10, the EOF is in t+1. The result for this example shows that as the forecast period draws nearer, the EOF decreases from 0.462, to -0.097, to 0.047, to 0.002. Using the average of the EOF absolute values as the MOE, Figure 9 presents the results expected. For each seasonal technique, there is an improvement as we forecast periods which are nearer to the present period.

SAMPLE EOF PROJECTIONS OUT FOUR PERIODS

t	t+1	t+2	t+3	t+4
5	0.06354	-0.02900	0.89280	-0.26160
6	0.00779	0.54802	-0.50548	0.11972
7	0.31060	0.23659	0.60275	**0.46215**
8	-0.24404	0.24630	**-0.09650**	0.56540
9	0.26845	**0.04728**	0.44275	-0.01953
10	**0.00171**	0.03610	-0.20234	0.08523
11	0.14652	0.02766	0.15643	-0.19487
12	0.07762	0.35258	0.07406	0.27510
13	0.03807	-0.31529	-0.12837	-0.27556
14	-0.20124	-0.11518	-0.27620	-0.44428
15	-0.02032	-0.25272	-0.40429	0.30848
16	-0.22514	-0.12923	0.23481	0.06135
17	-0.16356	-0.09883	-0.29731	-0.30556
18	0.08582	-0.12510	-0.16466	-0.15071
19	-0.19320	-0.22819	-0.24045	0.04632
20	-0.05956	0.24759	0.41768	0.13648
21	0.00524	0.08994	-0.12723	-0.10115
22	0.06678	-0.25627	-0.27049	-0.16744

Figure 8. Sample EOF Projected Four Periods

Average Absolute EOF Forecasting Out Four Periods				
Technique	t+1	t+2	t+3	t+4
Winters	0.1633	0.1894	0.2142	0.2373
Harrison-Stevens	0.1382	0.1471	0.1601	0.1621

Figure 9. Improved Forecasts in Nearer Periods

A comparative analysis between the Winters and Harrison-Stevens seasonality techniques is made for periods t+2, t+3, and t+4. Figures 10 through 12 graphically display the increasing superiority of the Harrison-Stevens technique when projecting further into the future. To verify this observation statistically, a oneway ANOVA test is made. The results shown in figure 13 indicate statistical significance between the two techiques for all three projections at the 90% confidence level. Additionally, we observe a strengthening of this significance as we project further out.

E. ERROR OF FORECAST ANALYSIS

An analysis of the EOF values is conducted to identify the presence of a distribution. Our theory is based upon the assumption that the error in forecasting is normally distributed. For future application of the techniques presented in this research, the normality assumption previously discussed is verified using a quantile plot of the Harrison-Stevens EOF data against the normal distribution. With a simple square root transformation (reattaching the

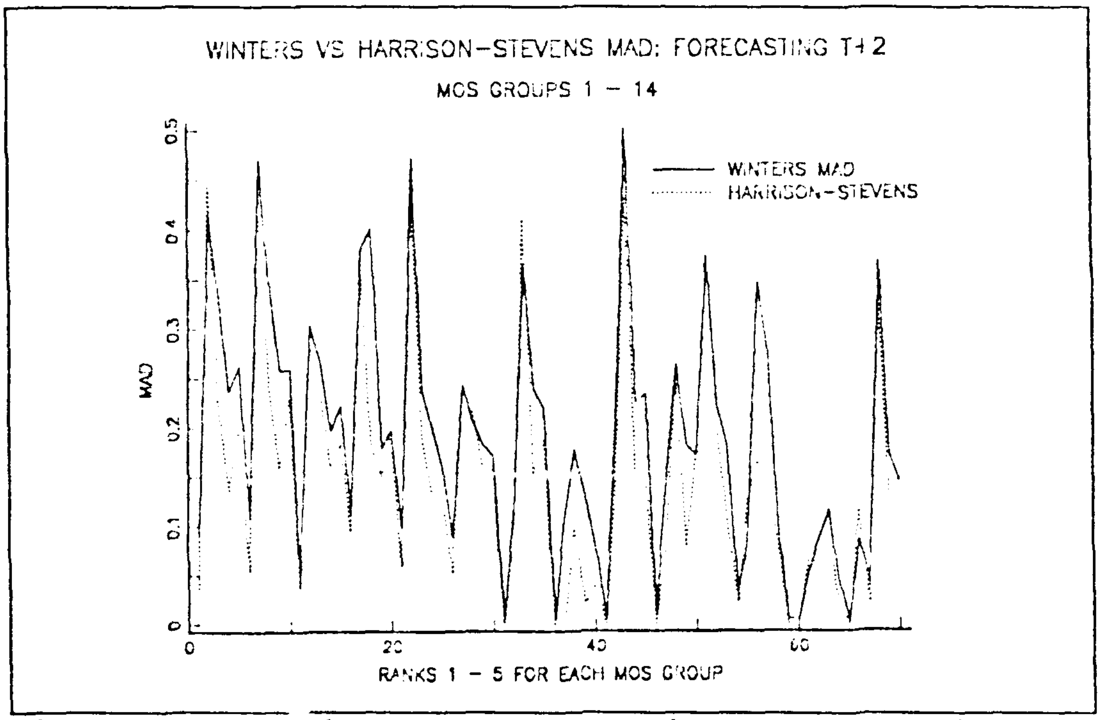

Figure 10. Harrison-Stevens vs. Winters EOF: period t+2

signs after transforming the EOF magnitudes), we obtain an excellent fit to the normal distribution.

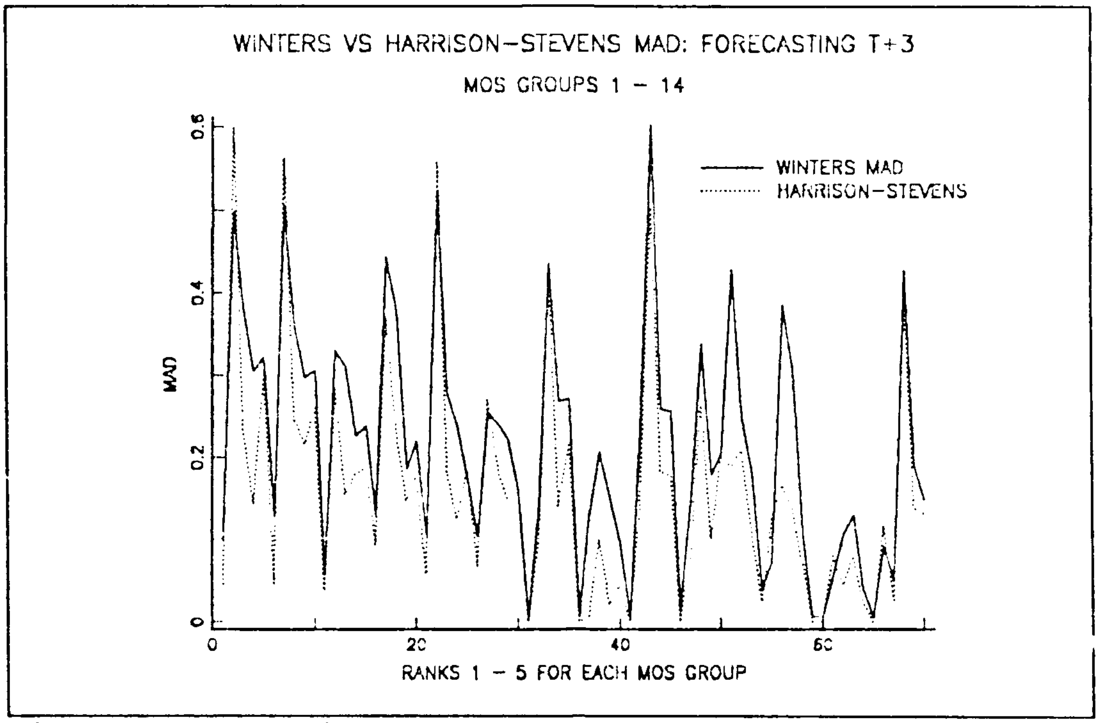

Figure 11. Harrison-Stevens vs Winters EOF: period t+3

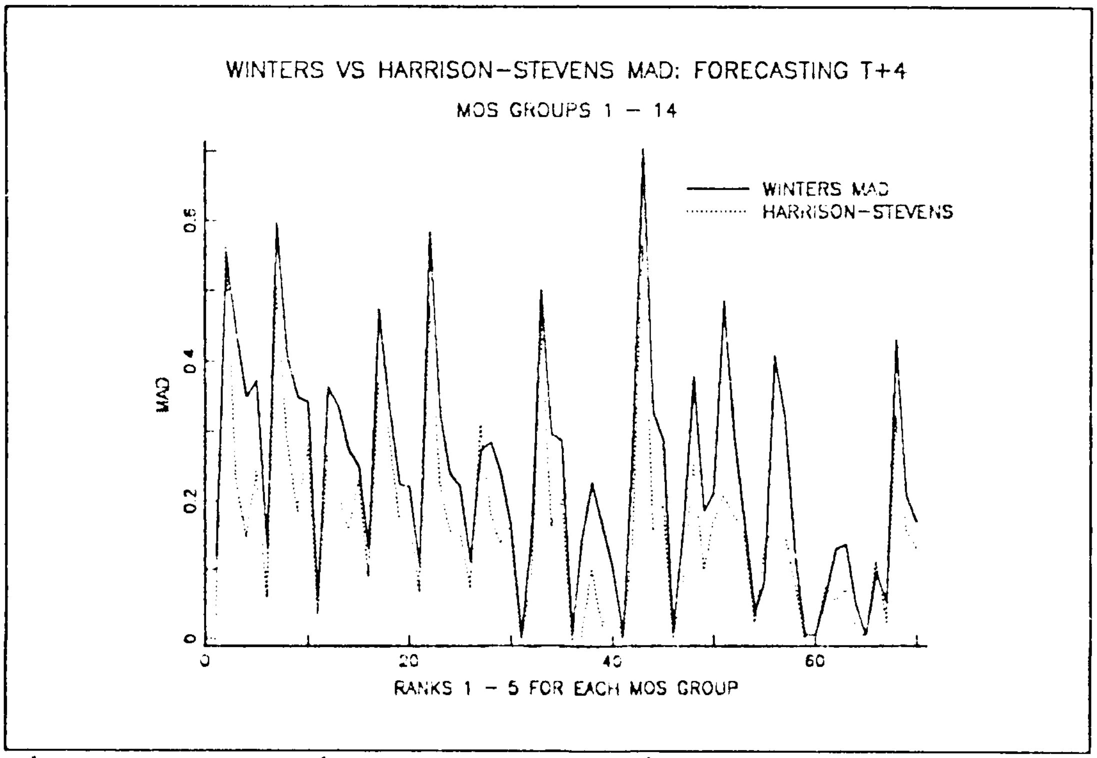

Figure 12. Harrison-Stevens vs Winters EOF: period t+4

ANOVA TABLE: WINTERS VS HARRISON-STEVENS (T + 2)

SOURCE	SUM OF SQR	DF	MEAN SQR	F
BETWEEN	0.062544721	1	0.062544721	4.371881748
WITHIN	1.974246338	138	0.014306133	
TOTAL	2.036791059	139		

ANOVA TABLE: WINTERS VS HARRISON-STEVENS (T + 3)

SOURCE	SUM OF SQR	DF	MEAN SQR	F
BETWEEN	0.102343967	1	0.102343967	5.185222873
WITHIN	2.723791788	138	0.019737622	
TOTAL	2.826135755	139		

ANOVA TABLE: WINTERS VS HARRISON-STEVENS (T + 4)

SOURCE	SUM OF SQR	DF	MEAN SQR	F
BETWEEN	0.198180206	1	0.198180206	9.027351471
WITHIN	3.029556181	138	0.021953306	
TOTAL	3.227736387	139		

Figure 13. ANOVA Comparison in Future Forecasts

VII. CONCLUSION

Majors Randy Larsen and John Misiewicz made significant strides in applying aggregation and shrinkage techniques for officer attrition rate estimations. While achieving estimation stability for the small cell problem, the results were based upon annual data and failed to consider seasonality as a factor. Our data allows us to successfully introduce the seasonality factor on a quarterly basis with flexibility of conversion to any cycle desired.

While we are grateful of Corporal Dean Hupp, MIIS, Headquarters, U.S. Marine Corps, and his efforts in preparing a usable data base, his resources were limited and many shortcomings exist which require sensitive manipulation. The available modified data base proves usable for model validation but would not serve well for actual forecasting. DSAI expects to have a quality data base by October 1990, and we recommend that it be used to verify our parameter estimations and modelling conclusions.

In general, the data base supports the use of seasonality factors for each MOS group/paygrade combination. How to incorporate seasonality into a model is open to debate. We recommend that the multiplicative approach be used and that a weighted update of the values be done at each period in the process.

The Winters Exponential Smoothing approach for estimating attrition rates is introduced to establish a baseline for the Harrison-Stevens approach. Our expectation of the Winters method was that it would present good forecasting results, but that it would not be competitive with the Harrison-Stevens technique. Using either MSE or MAD, forecasting is improved when seasonality is used. Further, the Harrison-Stevens approach yields better forecasts than the Winters method. While differences in the one-period forecasts are not statistically significant, we find this significance strengthened with each subsequent period estimated. Each method is capable of forecasting as far into the future as desired, and when forecasting out two or more periods, Harrison-Stevens is statistically superior to the Winters method. In addition, Harrison-Stevens provides a posterior distribution rather than a point estimate. It is recommended that the Harrison-Stevens Seasonal Multi-State Bayesian approach be used as the base model in the Marine Corps Officer Rate Generator.

Finally, we recommend that additional sensitivity analysis be conducted on remaining parameters used in the Harrison-Stevens model when a more sound data base is available.

APPENDIX A

A. SAMPLE DATA ENTRY

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
7	8	0	6		1	9		0	4		1	3	0	2		1					0					1

<u>Column</u>

1 - 2 Calendar Year

3 - 4 Ending Month of Observed Quarter

6 - 7 Years Commissioned Service (YCS)

9 - 10 Paygrade (O1 = 2nd Lt, O2 = 1st Lt, etc.)

12 - 15 Primary MOS (Actual USMC Codes)

17 Service Component (Regular Commission = 1 or Reserve Commission = 2)

19 - 22 Number of Attritions this Quarter

24 - 27 Ending Inventory this Quarter

B. SAMPLE SOURCE DATA BASE

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33...
7	5	6	2		3		1	0		1			8			9	4			4			9	6			3		1	0	3	

<u>Column</u>

1 - 4	Primary MOS
6	Paygrade
8 - 9	YCS
11	Service Component
12 - 14	Attrition for 1st Quarter 1978
14 - 18	Inventory for 1st Quarter 1978
19 - 21	Attrition for 2nd Quarter 1978
22 - 25	Inventory for 2nd Quarter 1978
26 - 28	Attrition for 3rd Quarter 1978
29 - 32	Inventory for 3rd Quarter 1978
. . .	. . .
47 - 49	Attrition for 2nd Quarter 1979
50 - 53	Inventory for 2nd Quarter 1979
. . .	. . .
114-117	Inventory for 4th Quarter 1989

APPENDIX B

A. FORTRAN PROGRAM: MCFIND

```
      SUBROUTINE MCFIND(MOS, SG,LG,MG)
C --- FIND LOCATION OF MATCHING MOS IN GROUP TABLE. RETURN GROUP NO
      PARAMETER (NMS=80, NG=14, NLG=6, NMG=4)
      INTEGER*2 MOSGR(2,NMS), LGRP(NG), MGRP(NLG)
      INTEGER SG,LG,MG
      COMMON /MOSTBL/ MOSGR, LGRP, MGRP
      DO 10 I=1,NMS
        IF(MOSGR(1,I) .EQ. MOS) THEN
                                   SG=MOSGR(2,I)
                                   LG=LGRP(SG)
                                   MG=MGRP(LG)
                                   RETURN
        ENDIF
   10 CONTINUE
      SG=0
      LG=0
      MG=0
C***  WRITE(6,*) '****** MOS NOT FOUND IN GROUP TABLE:',MOS
      END
C
      SUBROUTINE MOSGET(IX, MOS,IG,LG,MG)
      PARAMETER (NMS=80, NG=14, NLG=6, NMG=4)
      INTEGER*2 MOSGR(2,NMS), LGRP(NG), MGRP(NLG)
      COMMON /MOSTBL/ MOSGR, LGRP, MGRP
      MOS=MOSGR(1,IX)
      IG=MOSGR(2,IX)
      LG=LGRP(IG)
      MG=MGRP(LG)
      END
C
      BLOCK DATA
      PARAMETER (NMS=80, NG=14, NLG=6, NMG=4)
      INTEGER*2 MOSGR(2,NMS), LGRP(NG), MGRP(NLG)
      COMMON /MOSTBL/ MOSGR, LGRP, MGRP
      DATA MOSGR/302,1, 802,2, 1302,2, 1802,2, 1803,2,
     *   108,3, 202,3, 2502,3, 2602,3,
     *   3415,4, 4002,4, 4302,4, 5803,4,
     *   402,5, 3002,5, 3060,5, 3502,5, 6002,5,
     *   7204,6, 7208,6, 7210,6, 7320,6,
     *   7501,7, 7511,7, 7522,7, 7542,7, 7543,7, 7545,7, 7576,7,
```

```
     *   7521,8, 7523,8,
     *   7556,9, 7557,9, 7562,9, 7564,9, 7565,9, 7566,9, 7587,9,
     *   7508,10, 7509,10, 7563,10, 7581,10, 7583,10, 7584,10,
     *   7585,10, 7586,10, 7588,10,
     *   101,11, 201,11, 301,11, 401,11, 801,11, 1301,11, 1801,11,
     *   2501,11, 2601,11, 3001,11, 3401,11, 3501,11, 4001,11, 4301,11,
     *   4401,11, 5801,11, 6001,11, 7201,11, 7301,11, 9901,11,
     *   7580,12, 7597,12, 7598,12, 7599,12,
     *   7500,13, 7510,13, 7520,13, 7540,13, 7550,13, 7560,13, 7575,13,
     *   4402,14       /
C
      DATA LGRP/1,1, 2,2,2,2, 3,3, 4,4, 5,5,5, 6 /
      DATA MGRP/1,         1,    2,    2,      3, 4 /
      END
```

B. FORTRAN PROGRAM: MCMATX

```
C --- PROGRAM TO CREATE 3-DIM MATRICES MOS X PG X QUARTER
C --- JUNE 1990      LCU  MARINE CORPS
C ---   PROVIDES SGI/SGL FOR ALL DESIGNATED GROUP OVER 48 PERIODS
C --- PARAMETER MXY MUST BE UPDATED TO REFLECT EXACT NO. YEARS OF DATA
C ---
      PARAMETER (MXY=48, NSG=14, NLG=6, NMG=4, NPG=5, NQ=4)
      INTEGER MOS,PG, YCS, SVC,  SG,LG,MG, GROUP
      INTEGER INV(MXY), LOSS(MXY)
      REAL SGI(48,NPG)
      REAL SGL(48,NPG)

      DO 3 GROUP = 1,NSG

        DO 6 I = 1,MXY
          DO 7 J = 1,NPG
            SGL(I,J) = 0
            SGI(I,J) = 0
    7     CONTINUE
    6   CONTINUE

C ---
      NR=0
      NYR=MXY/4
      DO 10  I=1,999999
    5  READ(1,100,END=999) PG,MOS,YCS,SVC,INV,LOSS
       NR=NR+1
       CALL MCFIND(MOS, SG,LG,MG)
       IF(SG.NE.GROUP) GO TO 5
C ---
C ---  SUMMARIZE FOR EACH QUARTER
C      MXY3=MXY-3
```

```
      DO 20 J=1,MXY
        SGI(J,PG)=SGI(J,PG) + INV(J)
        SGL(J,PG)=SGL(J,PG) + LOSS(J)
   20  CONTINUE
   10 CONTINUE
C ---
  999 CONTINUE
      DO 40 J=1,NPG
       DO 50 K=1,48
C******  IF(SGI(K,J).EQ 0.) SGI(K,J)=1
   50  CONTINUE
   40 CONTINUE
C ---
C --- WRITE MATRICES OUT AS 2-DIM MATRICES ONE FOR EACH QTR

      WRITE((10+GROUP),202) GROUP
      WRITE((30+GROUP),202) GROUP
      DO 200 K=1,48
       WRITE((10+GROUP),201) (SGL(K,J),J=1,NPG)
       WRITE((30+GROUP),201) (SGI(K,J),J=1,NPG)
 200  CONTINUE
      WRITE(6,*) ' **** RECORD READ=',NR
 201  FORMAT(5F10.0)
 202  FORMAT(//,3X,'DATA FOR GROUP ',I3,/)
 100  FORMAT(I1,I4,I2,I1,96I4)
      REWIND(1)
    3 CONTINUE
      END
```

C. APL PROGRAM: OUTLY

```
        ∇ OUTLY X;X;ORDER;MED;IQR
[1]     ORDER←X[⍋X]
[2]     MED←(ORDER[(ρX)÷2]+ORDER[((ρX)÷2)+1])÷2
[3]     IQR←ORDER[(ρX)×0.75]-ORDER[(ρX)÷4]
[4]     SEE←(((IQR×3)⌈|(X-MED))-(IQR×3))
[5]     WHERE←(SEE≠0)/ιρSEE
[6]     ORDER
[7]     ' '
[8]     SEE
[9]     ' '
[10]    WHERE
        ∇
```

D. FORTRAN PROGRAM: MCFX

```
C --- PROGRAM TO FIX DATA BASE MC90C BY CHANGING CERTAIN VALUES
C --- TO THE AVERAGE OF 4 QTRS BEFORE AND AFTER.
```

```
C ---
C --- JULY 1990      LCU  MARINE CORPS
C --- PARAMETER MXY MUST BE UPDATED TO REFLECT EXACT NO. QTRS OF DATA
C ---
      PARAMETER (MXY=48, NSG=14, NPG=5, NQF=4, SGSIZE=NSG*MXY*NPG)
      INTEGER MOS,PG, YCS, SVC,  SG
      INTEGER INV(MXY), LOSS(MXY)
      INTEGER QT(NPG,NQF)
      DATA QT/45,45,12,8,14,    0,0,14,12,45,    0,0,45,14,0,
     *        0,0,0,45,0 /
C
C --- FIX INDIVIDUAL RECORDS ON UNAGGREGATED DATA BASE
      CALL FIXREC(QT,INV,LOSS,NPG,NSG,NQF, MXY)
      END
C
      SUBROUTINE FIXREC(QT,INV,LOSS,NPG,NSG,NQF, MXY)
C --- FIX EACH RECORD FROM SOURCE DATA BASE
      INTEGER MOS,PG, YCS, SVC,  SG
      INTEGER INV(MXY), LOSS(MXY), QT(NPG,NQF)
      REWIND(1)
      DO 10  I=1,999999
   5   READ(1,100,END=999) PG,MOS,YCS,SVC,INV,LOSS
       IF(PG.LT.1 .OR. PG.GT.5) GO TO 10
       CALL MCFIND(MOS, SG,LG,MG)
       IF(SG.LE.0) GO TO 10
C ---  FIX TIME SERIES
       DO 20 K=1,NQF
        IQ=QT(PG,K)
        IF(IQ.GT.0) THEN
            I1=IQ-4
            I2=IQ+4
            IF(I2.GT.MXY) I2=IQ+3
            INV(IQ)= .5*(INV(I1)+INV(I2)) + .5
            LOSS(IQ)= .5*(LOSS(I1)+LOSS(I2)) + .5
        ENDIF
  20   CONTINUE
       WRITE(2,100) PG,MOS,YCS,SVC,INV,LOSS
  10  CONTINUE
C ---
 999  CONTINUE
 100  FORMAT(I1,I4,I2,I1,96I4)
      END
```

E. FORTRAN PROGRAM: MC90

```
C --- PROGRAM TO ANALYZE MARINE CORPS PERSONNEL INVENTORY
C --- AND ATTRITION DATA.
```

```
C --- SEP 1987 REVISED FOR NEW DATA BASE FORMAT BY L. URIBE
C --- MAY 1989 REVISED FOR AGGREGATION ALGORITHMS  L. URIBE
C --- MAY 1990 REVISED FOR NEW DATA FORMAT          L. URIBE
C --- PARAMETER MXY MUST BE UPDATED TO REFLECT EXACT NO. OF QTRS
C ---
      PARAMETER (MXX=200, MXY=48, MXR=2000)
C ---
      INTEGER SYCS(31), NYCS
      INTEGER SYCSG(31),SYCSL(31),SYCSM(31),NYCSG,NYCSL,NYCSM
      INTEGER SMOS(80), NMOS
      INTEGER SVCMP(5), NSC
      INTEGER SGRD
      INTEGER*2 VYC(50)
C ---
      REAL INV(MXX,MXY),Y(MXX,MXY), SINV(MXX,MXY),SY(MXX,MXY)
      INTEGER DATA(MXY)
      REAL XTB(MXX),VXTB(MXX),XEB(MXX),A(MXX)
      INTEGER*2 PTRTBL(MXX, 2),INDX(MXX), MKG(MXX), RETTBL(MXR,5)
      INTEGER*2 PTBL(MXX, 3), BKTBL(MXX,3)
      REAL AVINV(MXX), RETINV(MXR)
      EQUIVALENCE (RETTBL, INV)
      DATA MKG/MXX*0/
C
      DO 1 I=1,MXX
       DO 2 J=1,MXY
        SINV(I,J)=0
          SY(I,J)=0
         INV(I,J)=0
           Y(I,J)=0
    2  CONTINUE
    1 CONTINUE
C
C --- INITIAL VALUE FOR AGGREGATION ESTIMATION PERCENTAGE
      AGGPCT=0.9
      ICYCLE=1
C
      CALL GETPAR(AIMIN,NO,NMOS,SMOS,NYCS,SYCS,SGRD,
     *            NSC,SVCMP,  IGR,LG,MG)
C --- MAJOR GROUP IS MG, LARGE GROUP LG, GROUP IGR, YCS BLOCK IY
              WRITE(6,*) ' '
              WRITE(6,*) '---- GR,LG,MG=',IGR,LG,MG
              WRITE(6,*) ' '
C --- READ EVAL TABLE. SELECT ONLY RECS PASSING SELECT CRITERIA


      CALL READET(RETTBL,RETINV,MXR,MXY,NRET,SGRD,NSC,SVCMP,MG)
    5 RC=0
      IGX=IGR
      LGX=0
      MGX=0
      NYCSG=1
```

```
      SYCSG(1)=SYCS(1)
      NYCSL=1
      SYCSL(1)=SYCS(1)
      NYCSM=1
      SYCSM(1)=SYCS(1)
      NCTOT=0
      NC=NCEVAL(AIMIN,IGX,LGX,MGX,NYCSG,SYCSG,RETTBL,RETINV,NRET,MXR,
     *                AGGPCT,IGR,LG)
C --- DO WHILE NCTOT<NO & RC=0    (EXPAND AS LONG AS NO NOT MET)
   10 IF(NC .GE. NO) THEN
        WRITE(6,*) '$GG EVAL NC,SYCSG=',NC,(SYCSG(II),II=1,NYCSG)
        GO TO 60
      ENDIF
      IF(NYCSG.EQ.1) THEN
        CALL GETVYC(SYCS(1),LG,NYE,VYC)
        WRITE(6,*) '=== VYC=',(VYC(I),I=1,NYE)
      ENDIF
      CALL EXPAND(NYCSG,SYCSG,VYC,NYE,IGX,LGX,MGX,LG,MG,RC)
      IF(IGX .EQ. 0) GO TO 20
      NC=NCEVAL(AIMIN,IGX,LGX,MGX,NYCSG,SYCSG,RETTBL,RETINV,NRET,MXR,
     *                AGGPCT,IGR,LG)
     GO TO 10
C
   20 NCTOT=NC
      WRITE(6,*) '$$G EVAL NC,SYCSG=',NCTOT,(SYCSG(II),II=1,NYCSG)
C --- EXPAND TO LARGE MOS GROUP
      WRITE(6,*) ' '
      WRITE(6,*) '=== EXPANDING BY LARGE GROUP:',LGX
      NC=NCEVAL(AIMIN,IGX,LGX,MGX,NYCSL,SYCSL,RETTBL,RETINV,NRET,MXR,
     *                AGGPCT,IGR,LG)
   30 IF((NCTOT+NC) .GE. NO) THEN
        WRITE(6,*)'$LL EVAL NC,SYCSL=',(NCTOT+NC),(SYCSL(II),II=1,NYCSL)
        GO TO 60
      ENDIF
      IF(NYCSL.EQ.1) CALL GETVYC(SYCS(1),LG,NYE,VYC)
      CALL EXPAND(NYCSL,SYCSL,VYC,NYE,IGX,LGX,MGX,LG,MG,RC)
      IF(LGX .EQ. 0) GO TO 40
      NC=NCEVAL(AIMIN,IGX,LGX,MGX,NYCSL,SYCSL,RETTBL,RETINV,NRET,MXR,
     *                AGGPCT,IGR,LG)
      GO TO 30
C
   40 NCTOT=NCTOT+NC
      WRITE(6,*) '$$L EVAL NC,SYCSL=',NCTOT,(SYCSL(II),II=1,NYCSL)
C --- EXPAND TO MAJOR MOS GROUP
      WRITE(6,*) ' '
      WRITE(6,*) '=== EXPANDING BY MAJOR GROUP:',MGX
      NC=NCEVAL(AIMIN,IGX,LGX,MGX,NYCSM,SYCSM,RETTBL,RETINV,NRET,MXR,
     *                AGGPCT,IGR,LG)
   50 IF((NCTOT+NC) .GE. NO  .OR.  RC .NE. 0) THEN
        WRITE(6,*)'$MM EVAL NC,SYCSM=',(NC+NCTOT),(SYCSM(II),II=1,NYCSM)
        GO TO 60
```

```
      ENDIF
      IF(NYCSM.EQ.1) CALL GETVYC(SYCS(1),LG,NYE,VYC)
      CALL EXPAND(NYCSM,SYCSM,VYC,NYE,IGX,LGX,MGX,LG,MG,RC)
      NC=NCEVAL(AIMIN,IGX,LGX,MGX,NYCSM,SYCSM,RETTBL,RETINV,NRET,MXR,
     *              AGGPCT,IGR,LG)
      GO TO 50
C
C --- EXPANSION FINISHED
   60 IF(RC .NE. 0) THEN
        WRITE(5,*)'*** REQUIRED NO MAY NOT BE MET: NO,NC=',NO,(NC+NCTOT)
      ENDIF
C
      WRITE(5,*) 'ESTIMATED NUMBER OF CELLS =',NC+NCTOT
   70 WRITE(5,*)
      WRITE(5,*) 'ENTER 1 TO CALL READER, 0 TO CHANGE EXPANSION'
      READ(5,*) NPICK1
      IF(NPICK1 .EQ. 1) THEN
        GO TO 80
      ELSE
        WRITE(5,*) 'AGGPCT IS CURRENTLY =', AGGPCT
        WRITE(5,*) 'ENTER NEW VALUE FOR AGGPCT'
        READ(5,*) AGGPCT
        GO TO 5
      ENDIF
   80 WRITE(5,*) 'CALLING READER'
C
      CALL GETMOS(SMOS,NMOS,MGX,LGX,MG,LG,IGR)
C
      CALL READER(INV,Y,MXX,MXY,NMOS,NYCSG,NYCSL,NYCSM,NSC,
     *  SMOS,SYCSG,SYCSL,SYCSM,SGRD,SVCMP,NRC,PTRTBL,LGX,MGX,
     *  IGR,LG,NPT,PTBL,SINV,SY)
C
      CALL AGGREG(INV,Y,MXX,MXY,SMOS,SYCSG,
     *      NRC, NRCOLD,PTRTBL,INDX,AVINV,AIMIN,MKG)
C
      WRITE(5,*) 'NUMBER OF CELLS =',NRC
   90 WRITE(5,*)
      WRITE(5,*) 'ENTER 1 TO CONTINUE, 0 TO CHANGE EXPANSION'
      READ(5,*) NPICK2
      IF(NPICK2 .EQ. 1) THEN
        GO TO 100
      ELSE
        WRITE(5,*) 'AGGPCT IS CURRENTLY =', AGGPCT
        WRITE(5,*) 'ENTER NEW VALUE FOR AGGPCT'
        READ(5,*) AGGPCT
        ICYCLE=ICYCLE+1
        GO TO 5
      ENDIF
C
  100 CONTINUE
      WRITE(6,201)'EXPANSION INFORMATION:'
```

```
      WRITE(6,203)'ACTUAL NO. OF CELLS USED= ',NRC
      WRITE(6,202)'MOS GROUP #',IGR,' YCS''S USED=',
     *             (SYCSG(I),I=1,NYCSG)
      IF(LGX .GT. 0) THEN
        WRITE(6,204)'LARGE MOS GROUP #',LG,' YCS''S USED=',
     *                (SYCSL(I),I=1,NYCSL)
      ELSE IF(MGX .GT. 0) THEN
        WRITE(6,204)'LARGE MOS GROUP #',LG,' YCS''S USED=',
     *                (SYCSL(I),I=1,NYCSL)
        WRITE(6,204)'MAJOR MOS GROUP #',MG.' YCS''S USED=',
     *                (SYCSM(I),I=1,NYCSM)
      ENDIF
C*********************************** ==================
                    STOP
C
C**** CALL MC87BZ(INV,Y,NRC,MXY,XTB,VXTB,XEB,A,MXX,MXY)
C
      CALL BKDOWN(PTBL,NPT,PTRTBL,NRCOLD,INDX,MKG,MXX,MXY,
     *             SINV,SY,INV,Y,BKTBL,NBK    )
C
 201  FORMAT(/1X,A)
 202  FORMAT(1X,A,I2,A/1X,18(I3))
 203  FORMAT(1X,A,I2)
 204  FORMAT(1X,A,I1,A/1X.18(I3))
      END
C
*****************************************************************
C
      SUBROUTINE EXPAND(NYCSX,SYCSX,VYC,NYE,IGX,LGX,MGX.LG,MG,RC)
C --- EXPAND YCS IF FEAS, ELSE EXP MOS TO LG/MG & BACKTRACK YCS
      INTEGER SYCSX(31), NYCSX
      INTEGER*2 VYC(NYE)
C --- FIND POSITION OF ORIGINALLY REQUESTED SYCS(1)
      IY=0
      DO 10 I=1,NYE
       IF(SYCSX(1) .EQ. VYC(I)) IY=I
  10  CONTINUE
      IF(IY.EQ.0) GO TO 30
C --- FIND NEAREST NON-ZERO YCS TO USE FOR EXPANSION
      DO 20 I=1,NYE
       J=IY-I
       IF(J.GE.1) THEN
            IF(VYC(J).GT.0) GO TO 50
       ENDIF
       J=IY+I
       IF(J.LE.NYE) THEN
            IF(VYC(J).GT.0) GO TO 50
       ENDIF
  20  CONTINUE
  30  CONTINUE
***         WRITE(6,*) '---- YCS EXPANSION FINISHED:  IY=',IY
```

```
C --- NO MORE YCS EXPANSION POSSIBLE. SEE IF MOS EXP. FEASIBLE
      IF(IGX.GT.0) THEN
C ---         EXPAND FROM GROUPS TO LARGE GROUP LGX. BACK YCS
              IGX=0
              LGX=LG
      ELSE IF(LGX.GT.0) THEN
C ---   EXPAND FROM LARGE GROUP LGX TO MAJOR GROUP MGX. BACK YCS
              LGX=0
              MGX=MG
      ELSE
              RC=1
      ENDIF
      RETURN
C
C --- EXPAND WITH YCS IN POSITION J & CLEAR VYC(J)
   50 CONTINUE
      NYCSX=NYCSX+1
      SYCSX(NYCSX)=VYC(J)
      VYC(J)=0
      END
C
*****************************************************************
C
      FUNCTION NCEVAL(AIMIN,IGX,LGX,MGX,NYCSX,SYCSX,RETTBL,RETINV,
     *     NRET,MXR,AGGPCT,IGR,LG)
C --- COMPUTE EST. NO. CELLS TO OBTAIN WITH CURRENT SELECTION
      INTEGER SYCSX(31),NYCSX
      INTEGER*2 LGRP(14),MGRP(6)
      INTEGER*2 RETTBL(MXR, 5)
      REAL RETINV(MXR)
      LOGICAL ACCEPT
      NCEVAL=0
      IF(IGX.EQ.0 .AND. LGX.EQ.0 .AND. MGX.EQ.0) RETURN
      TAINV=0.0
      DO 100 I=1,NRET
C ---  SCREEN ON YCS
       DO 10 J=1,NYCSX
        IF(RETTBL(I,2) .EQ. SYCSX(J)) GO TO 15
   10  CONTINUE
       GO TO 100
C ---  SCREEN ON MOS BY GROUP, LG/MG DEPENDS ON IGX,LGX,MGX
   15  CONTINUE
       MOS=RETTBL(I,1)
       IGP=RETTBL(I,3)
       LGP=RETTBL(I,4)
       MGP=RETTBL(I,5)
       ACCEPT=.FALSE.
       IF(MGX .GT. 0) THEN
        IF(MGP .EQ. MGX .AND. LGP .NE. LG) ACCEPT=.TRUE.
       ELSE
       IF(LGX .GT. 0) THEN
```

```
        IF(LGP .EQ. LGX .AND. IGP .NE. IGR) ACCEPT=.TRUE.
       ELSE
       IF(IGP .EQ. IGX) ACCEPT=.TRUE.
       ENDIF
C ---  ACCEPTED
       IF(ACCEPT) THEN
        IF(RETINV(I) .GE. AIMIN) THEN
         IF(MGX.GT.0) WRITE(6,*) 'NCEVAL,MOS,YCS,IGP,LGP,MGP=',
     *    NCEVAL,MOS,RETTBL(I,2),IGP,LGP,MGP
              NCEVAL=NCEVAL+1
        ELSE
              TAINV=TAINV+RETINV(I)
        ENDIF
       ENDIF
  100 CONTINUE
C
C --- FINAL ESTIMATE IS NCEVAL
      IF(AIMIN.GT.0) NCEVAL=NCEVAL + AGGPCT*TAINV/AIMIN
       WRITE(6,*) 'NCEVAL,TAINV,IGX,LGX,MGX=',NCEVAL,TAINV,IGX,LGX,MGX
      END
C
*****************************************************************
C
      SUBROUTINE GETVYC(SYCS, LG, NY,VYC)
      PARAMETER (NYB=4, NYE=18, NYEG=4)
      INTEGER*2 YCSB(NYE,NYB,NYEG), VYC(NYE), LGEX(6)
      INTEGER SYCS
      DATA LGEX/4,4,4,2,1,3/
      DATA YCSB/1,2,3,4,5,6, 8,9,10,11,12,13,14,15,16,17,18,19,
     *         7,17*0,    20,21,22,23,24,25,12*0,    26,17*0,
     *         1,2,3,4,5, 8,9,10,11,12,13,14,15,16,17,18,19,1*0,
     *         6,7,16*0,    20,21,22,23,24,25,12*0,    26,17*0,
     *         1,2,3,4,5,6, 8,9,10,11,12,13,14,15,16,17,18,19,
     *         7,17*0,    20,21,22,23,24,25,12*0,    26,17*0,
     *         1,2,3, 6,7,8,9,10,11,12,13,14,15,16,17,18,19,1*0,
     *         4,5,16*0,    20,21,22,23,24,25,12*0,    26,17*0    /
C --- L INDICATES LAST DIMENSION IN YCS EXPANSION TABLE
      L=LGEX(LG)
      NY=NYE
C --- FIND TO WHICH YCS BLOCK SYCS BELONGS AND MAKE COPY IN VYC
      DO 10 J=1,NYB
       DO 20 I=1,NYE
        IF(SYCS .EQ. YCSB(I,J,L)) THEN
              DO 30 K=1,NYE
              VYC(K)=YCSB(K,J,L)
   30         CONTINUE
              RETURN
        ENDIF
   20  CONTINUE
   10 CONTINUE
      WRITE(6,*) '****** Y [illegible] IN YCSB TABLE YCS=',SY
```

```
      END
C
*********************************************************************
C
      SUBROUTINE READET(RETTBL,RETINV,MXR,MXY,NRET,SGRD, NSC,SVCMP, MG)
      INTEGER INVENT(100), LEAV(100)
C --- READ TABLE WITH ALL EXISTING COMBINATIONS FOR SELECTION CRITERIA
C --- ACCEPT RECS WITH MATCHING PG,MG,SVC.  ACCEPT ALL YCS
      INTEGER SVCMP(5), NSC, SVC
      INTEGER SGRD, PG
      INTEGER MOS,YCS
      REAL RETINV(MXR), AI
      INTEGER*2 RETTBL(MXR, 5)
      NRET=0
      DO 10 I=1,999999
      READ(1,101,END=999) PG,MOS,YCS,SVC, (INVENT(K),K=1,MXY),
     *                                    (LEAV(K),K=1,MXY)
       IF(PG.LT.SGRD) GO TO 10
       IF(PG.GT.SGRD) GO TO 999
       CALL MCFIND(MOS, IGR,LG,MGX)
       IF(MGX .NE. MG) GO TO 10
       DO 20 J=1,NSC
        IF(SVC .EQ. SVCMP(J)) THEN
           NRET=NRET+1
           IF(NRET.GT.MXR) STOP 555
           RETTBL(NRET,1)=MOS
           RETTBL(NRET,2)=YCS
           RETTBL(NRET,3)=IGR
           RETTBL(NRET,4)=LG
           RETTBL(NRET,5)=MG
           RETINV(NRET)=AVG(INVENT,MXY)
C***       WRITE(6,104) NRET,MOS,YCS.IGR,LG,MG,RETINV(NRET)
           GO TO 10
        ENDIF
   20  CONTINUE
   10 CONTINUE
  999 CONTINUE
  101 FORMAT(I1,I4,I2,I1,200I4)
  104 FORMAT(6I6, F9.2)
      END
C
*********************************************************************
      FUNCTION AVG(IV,N)
      INTEGER IV(N)
      AVG=0.
      DO 10 I=1,N
       AVG=AVG+IV(I)
   10 CONTINUE
      AVG=AVG/N
      END
```

```
      SUBROUTINE GETPAR(AIMIN,NO,NMOS,SMOS,NYCS,SYCS,SGRD,
     *               NSC,SVCMP,   IGR,LG,MG)
C --- GET SELECTION CRITERIA FROM USER AND VALIDATE
      INTEGER SYCS(31), NYCS
      INTEGER SMOS(80), NMOS
      INTEGER SVCMP(5), NSC
      INTEGER SGRD
      WRITE(5,*) ' ENTER THRESHOLD MIN. FOR AVERAGE INVENTORY'
      READ(5,*) AIMIN
      WRITE(5,*) ' ENTER THRESHOLD MIN. FOR NUMBER OF CELLS'
      READ(5,*) NO
      WRITE(5,*) ' THRESHOLDS TO USE AIMIN, NO=',AIMIN,NO
C
      WRITE(5,*) ' ENTER MOS (ONLY 1 ACCEPTED)'
      NMOS=1
      READ(5,*) SMOS(1)
      WRITE(6,*) ' MOS SELECTED:', SMOS(1)
      CALL MCFIND(SMOS(1), IGR,LG,MG)
      WRITE(6,*) ' GROUP TO USE:', IGR
      IF(IGR.EQ.0) THEN
               WRITE(5,*) '**** ERROR - INVALID MOS SELECTED:',SMOS(1)
               STOP
      ENDIF
C
      WRITE(5,*) ' ENTER YCS (ONLY 1 ACCEPTED)'
      NYCS=1
      READ(5,*) SYCS(1)
      WRITE(6,*) ' YCS SELECTED:', SYCS(1)
C
      WRITE(5,*) ' ENTER GRADE'
      READ(5,*) SGRD
      WRITE(6,*) ' GRADE SELECTED', SGRD
C
      WRITE(5,*) ' ENTER NO. OF SVC. COMPS & ARRAY (1-3, 4=1+2, 5=ALL)'
      READ(5,*) NSC, (SVCMP(I), I=1,NSC)
C --- EXPAND 4 TO 1,2 AND 5 TO 1,2,3
      DO 10 I=1,NSC
       IF(SVCMP(I).EQ.4 .OR. SVCMP(I).EQ.5) THEN
          NSC=SVCMP(I)-2
          DO 15 J=1,NSC
           SVCMP(J)=J
   15     CONTINUE
          GO TO 11
       ENDIF
   10 CONTINUE
   11 CONTINUE
      WRITE(6,*) ' SERVICE COMPONENTS SELECTED', (SVCMP(I), I=1,NSC)
C
      WRITE(6,101) 'TEST CASE INPUT PARAMETERS:'
      WRITE(6,102) 'INVENTORY THRESHOLD= ',AIMIN,
     *              'THRESHOLD NO. OF CELLS= ',NO
```

```
      WRITE(6,103) 'MOS= ',SMOS(1),'YCS= ',SYCS(1),'GRADE= ',SGRD
      WRITE(6,104) 'SERVICE COMPONENTS= ',(SVCMP(I),I=1,NSC)
C
 101  FORMAT(1X,A)
 102  FORMAT(1X,A,F4.1,7X,A,I2)
 103  FORMAT(1X,A,I5,2(5X,A,I2))
 104  FORMAT(1X,A,15(I3))
      END
C
***********************************************************************
C
      SUBROUTINE GETMOS(SMOS,NMOS, MGX,LGX, MG,LG,IG)
C --- BUILD SMOS ARRAY BASED UPON EXPANSION
      INTEGER SMOS(80)
      NMOS=0
      DO 10  I=1,999999
       CALL MOSGET(I, MOS,IGP,LGP,MGP)
       IF(MOS.LE.0) RETURN
       IF(MGX.GT.0 .AND. MGP.EQ.MG  .OR.
     *     LGX.GT.0 .AND. LGP.EQ.LG  .OR.
     *                    IGP.EQ.IG ) THEN
                           NMOS=NMOS+1
                           SMOS(NMOS)=MOS
       ENDIF
  10  CONTINUE
      END
C
***********************************************************************
C
      SUBROUTINE READER(INV,Y,MXX,MXY,NMOS,NYCSG,NYCSL,NYCSM,NSC,
     * SMOS,SYCSG,SYCSL,SYCSM,SGRD,SVCMP,NRC,PTRTBL,LGX,MGX,
     * IGR,LG, NPT,PTBL, SINV,SY)
      REAL INV(MXX,MXY),Y(MXX,MXY), SINV(MXX,MXY),SY(MXX,MXY)
      INTEGER INVENT(100), LEAV(100)
      INTEGER*2 PTRTBL(MXX, 2), PTBL(MXX,3)
      INTEGER SYCSG(*), SYCSL(*), SYCSM(*)
      INTEGER SMOS(*), NMOS
      INTEGER SVCMP(*), NSC
      INTEGER SGRD
      INTEGER YCS,PG,MOS,SVC
C
      REWIND 1
      DO 6  I=1,MXX
       DO 5  J=1,MXY
        INV(I,J)=0.0
          Y(I,J)=0.0
        SINV(I,J)=0.0
          SY(I,J)=0.0
  5    CONTINUE
  6   CONTINUE
C ---
```

```
      ICR=0
      NRC=0
      NPT=0
      ICNT=0
C
    1 READ(1,101,END=999) PG,MOS,YCS,SVC, (INVENT(I),I=1,MXY),
     *                                    (LEAV(I),I=1,MXY)
      ICR=ICR+1
C --- CHECK IF RECORD MEETS SELECTION CRITERIA.  OTHERWISE REJECT.
C
      IF(PG .LT. SGRD) GO TO 1
      IF(PG .GT. SGRD) GO TO 999
C
      CALL MCFIND(MOS, IGP,LGP,MGP)
      IF(IGP.EQ.0) GO TO 1
C
      IY=0
      IF(MGX .GT. 0) THEN
             IF(LGP .EQ. LG) THEN
                          CALL CKTBL(YCS,NYCSL,SYCSL,IY)
             ELSE IF(MGP .EQ. MGX) THEN
                          CALL CKTBL(YCS,NYCSM,SYCSM,IY)
             ENDIF
      ELSE IF(LGX .GT. 0) THEN
             IF(IGP .EQ. IGR) THEN
                          CALL CKTBL(YCS,NYCSG,SYCSG,IY)
             ELSE IF(LGP .EQ. LGX) THEN
                          CALL CKTBL(YCS,NYCSL,SYCSL,IY)
             ENDIF
      ELSE
        IF(IGP .EQ. IGR)   CALL CKTBL(YCS,NYCSG,SYCSG,IY)
      ENDIF
      IF(IY.EQ.0) GO TO 1
C
      CALL CKTBL(MOS,NMOS,SMOS,IM)
      IF(IM.EQ.0)  THEN
          WRITE(6,*) '*** ERROR IN MOS SCREENING ***',MOS
          WRITE(6,*) 'NMOS,SMOS=',NMOS,(SMOS(I),I=1,NMOS)
          GO TO 1
      ENDIF
C
      CALL CKTBL(SVC,NSC,SVCMP,IS)
      IF(IS.EQ.0) GO TO 1
C
C --- RECORD ACCEPTED - INSTALL IN INV,Y,SINV,SY, PTRTBL AND PTBL
      ICNT=ICNT+1
      IW=IS
      MINV=GINV(PTRTBL, MXX,NRC, IM,IY,-99)
      MV=GINV(PTBL, MXX,NPT,IM,IY,IW)
      CALL INSINV(PTRTBL,MXX,MXY,NRC,MINV,IM,IY,-99,INV,INVENT)
      CALL INSINV(PTBL,  MXX,MXY,NPT,MV,   IM,IY, IW,SINV,INVENT)
```

```
      CALL INSY(MXX,MXY,MINV,Y,LEAV)
      CALL INSY(MXX,MXY,MV,  SY,LEAV)

      GO TO 1
C
  999 CONTINUE
      WRITE(6,*) ' '
      WRITE(6,*) 'TOTAL RECORDS READ                =',ICR
      WRITE(6,*) 'TOTAL INV. MOS/YCS    COMBINATIONS=',NRC
      WRITE(6,*) 'TOTAL INV. MOS/YCS/IW COMBINATIONS=',NPT
      WRITE(6,*) 'TOTAL RECORDS ACCEPTED            =',ICNT
C --- TERMINATE IF NO DATA COLLECTED
      IF(NRC .EQ. 0) THEN
               WRITE(6,*) '**** NO DATA MEETS SELECTION REQS'
               STOP
      ENDIF
C
      WRITE(6,*) '     **** PTRTBL TABLE ****'
      DO 200 I=1,NRC
       WRITE(6,131)  I,(PTRTBL(I,J),J=1,2)
  200 CONTINUE
      WRITE(6,*) '     **** PTTBL  TABLE ****'
      WRITE(6,132) (I,(PTBL(I,J),J=1,3),(SINV(I,J),J=1, 10),I=1,NPT)
C
 101  FORMAT(I1,I4,I2,I1,200I4)
 121  FORMAT(A8,13I6)
 122  FORMAT(A8,7I6, 5X, 12I6)
 131  FORMAT(I4, 2I6)
 132  FORMAT(I4, 3I6, 10F7.2)
      END
C
      SUBROUTINE CKTBL(SRC,NTBL,TBL,IX)
      INTEGER TBL(NTBL), SRC
      DO 10  I=1,NTBL
       IF(SRC .EQ. TBL(I)) THEN
              IX=I
              RETURN
       ENDIF
  10  CONTINUE
      IX=0
      END
*************************************************************************
C
      FUNCTION GINV(PTBL, MXX,NPT, IM,IY,IW)
C --- FIND LOCATION OF INVENTORY ENTRY FOR MOS,YCS,SVC COMBINATIONS
C --- 3RD DIMENSION CHECKED ONLY IN CASE IW>0
      INTEGER*2 PTBL(MXX, *)
      DO 10 I=1,NPT
        IF(PTBL(I, 1) .EQ. IM  .AND.
     *     PTBL(I, 2) .EQ. IY ) THEN
           IF(IW.LT.0 .OR. (IW.GT.0 .AND. PTBL(I, 3).EQ.IW)) THEN
```

```
                                                 GINV=I
                                                 RETURN
            ENDIF
        ENDIF
   10 CONTINUE
      GINV=0
      END
C
      SUBROUTINE INSINV(PT,MXX,MXY,N,K,IM,IY,IW,INV,DATA)
C --- ACCUMM INTO KTH ENTRY.  INSTALL IN POINTER TABLE IF NOT PRESENT
      REAL INV(MXX, MXY)
      INTEGER*2 PT(MXX, *)
      INTEGER DATA(MXY)
      IF(K .EQ. 0) THEN
C ---       ADD NEW ENTRY
            N=N+1
            IF(N .GT. MXX) THEN
              WRITE(6,*) '*** ERROR - TOO MANY INV. COMBINATIONS',N
              STOP
            ENDIF
            K=N
            PT(K, 1)=IM
            PT(K, 2)=IY
            IF(IW.GT.0) PT(K, 3)=IW
      ENDIF
      DO 130 IT=1,MXY
       INV(K,IT)=INV(K,IT) + DATA(IT)
 130  CONTINUE
      END
***************************************************************************
C
      SUBROUTINE INSY(MXX,MXY,K,Y,DATA)
C --- ACCUMM INTO KTH ENTRY FOR LOSS
      REAL Y(MXX, MXY)
      INTEGER DATA(MXY)
      IF(K .EQ. 0) RETURN
      DO 10 IT=1,MXY
       Y(K,IT)=Y(K,IT) + DATA(IT)
   10 CONTINUE
      END
***************************************************************************
C
      SUBROUTINE AGGREG(INV,Y,MXX,MXY,SMOS,SYCSG,
     *                 NRC,NRCOLD,PTRTBL,INDX,AVINV, AIMIN,MKG)
C --- COMP. AVERAGE INV. & SORT
      REAL INV(MXX, MXY), Y(MXX, MXY), AVINV(MXX)
      INTEGER*2 PTRTBL(MXX, 2), INDX(MXX),MKG(MXX)
      INTEGER SYCSG(*), SMOS(*)
      REAL*8 TINV,TY
C
C --- RESET MKG (NECESSARY WHEN CYCLING THRU AGGPCT VALUES)
```

```
      DO 10  I=1,MXX
        MKG(I)=0
 10   CONTINUE
      TINV=0
      TY=0
      DO 100 I=1,NRC
C ---  FIX INV. ENTRIES LOWER THAN CORRESP. LOSSES & COMP. AVG INV.
       AI=0
       DO 201 J=1,MXY
        IF(INV(I,J).LT.Y(I,J)) INV(I,J)=Y(I,J)
        AI=AI+INV(I,J)
        TINV=TINV+INV(I,J)
          TY=  TY+  Y(I,J)
  201  CONTINUE
       AVINV(I)=AI/MXY
       INDX(I)=I
  100 CONTINUE
      WRITE(6,*) '===== TOTAL INV,Y=',TINV,TY
C
C --- SORT ASCENDING BY  AVG INVENTORY
      CALL SORT2(AVINV,INDX,NRC)
C
      NS1=0
C --- DISPLAY TABLE IN SORT SEQUENCE
      CALL DSPTBL(INV,Y,AVINV,PTRTBL,INDX,AIMIN,NRC,MKG,MXX,MXY.
     *                    SYCSG,SMOS )
C
      DO 200 K=NRC,1,-1
       IF(AVINV(K) .GE. AIMIN) THEN
C ---              MARK AS MEMBER OF SET S0
                   MKG(K)=32767
       ELSE
C ---              INITIAL COUNT OF MEMBERS OF SET S1
                   NS1=K
                   GO TO 202
       ENDIF
  200 CONTINUE
  202 CONTINUE
C --- DO AGGREGATIONS WITHIN SET S1 UNTIL NO MORE POSSIBLE (KF GE 0)
      KF=-1
C --- DO WHILE KF<0
  300 IF(KF.GE.0) GO TO 310
       CALL AGG1(AVINV,INDX,MKG,NS1,INV,Y,MXX,MXY,AIMIN,KF)
       GO TO 300
  310 CONTINUE
C --- DISPLAY TABLE AFTER 1ST AGGREGATION
***   CALL DSPTBL(INV,Y,AVINV,PTRTBL,INDX.AIMIN,NRC,MKG,MXX,MXY,
***  *                    SYCSG,SMOS )
      IF(NS1.EQ.NRC) THEN
           WRITE(6,*) '***** SET S0 EMPTY. NO CELLS ABOVE THRESHOLD'
           STOP
```

```
      ENDIF
C --- DO AGGREGATIONS FROM SET S1 INTO SET S0 UNTIL NO MORE POSSIBLE
      KF=1
C --- DO WHILE KF>0
  320 IF(KF.LE.0) GO TO 330
         CALL AGG2(AVINV,INDX,MKG,NS1,NRC,INV,Y,MXX,MXY, KF)
         GO TO 320
  330 CONTINUE
C --- DISPLAY TABLE AFTER 2ND AGGREGATION
      CALL DSPTBL(INV,Y,AVINV,PTRTBL,INDX,AIMIN,NRC,MKG,MXX,MXY,
     *                  SYCSG,SMOS )
C --- MOVE VALUES GE AIMIN TO BEGINNING OF ARRAYS
      CALL CMPRS(INV,Y,MXX,MXY,NRC,NRCOLD,AIMIN,AVINV)
C --- DISPLAY TABLE AFTER MOVING VALUES.
      DO 400 K=1,NRC
       WRITE(6,122)K,AVINV(K), (INV(K,J),J=1,12 )
       WRITE(6,123)             (  Y(K,J),J=1,12 )
  400 CONTINUE
  122 FORMAT(/I5,14X,F8.3, 6X,     12F7.2)
  123 FORMAT(     33X, 12F7.2)
      END
C ***************
      SUBROUTINE AGG1(AVINV,INDX,MKG,NS1,INV,Y,MXX,MXY,AIMIN,KF)
C --- DO ONE PASS OF AGGREGATION
      REAL INV(MXX, MXY), Y(MXX, MXY), AVINV(MXX)
      INTEGER*2 INDX(MXX),MKG(MXX)
      KF=0
      CI=0
      DO 10 I=NS1,1,-1
       IF(MKG(I).EQ.0) THEN
         IF(KF.EQ.0) THEN
C ---        THIS WILL BE THE COLLECTING CELL
            KF=I
            CI=AVINV(I)
         ELSE
            IF(CI+AVINV(I).LT.AIMIN) THEN
C ---           ACCUM. WITH CELL KF TEMPORARILY. SET TEMP. POINTER -KF
                CI=CI+AVINV(I)
                MKG(I)=-KF
            ELSE
C ---           FIND SMALLEST CELL TO ADD
                CALL AGG1A(AVINV,MKG,I,CI,AIMIN,KF,MXX)
            ENDIF
            IF(CI.GE.AIMIN) THEN
C ---           MAKE THIS AGGREGATION PERMANENT AND EXIT
                AVINV(KF)=CI
                CALL AGG1B(INDX,MKG,KF,INV,Y,MXY,MXX)
                NS1=NS1-1
                MKG(KF)=32767
                KF=-1
                RETURN
```

```
            ENDIF
          ENDIF
        ENDIF
   10 CONTINUE
C
      IF(KF.EQ.0) RETURN
C --- CLEAR TEMPORARY POINTERS LEFT.  THIS WAS AN UNSUCCESSFUL AGGREG.
      DO 20 I=1,NS1
       IF(MKG(I).LT.0) MKG(I)=0
   20 CONTINUE
      END
C ***************
      SUBROUTINE AGG1A(AVINV,MKG,ILAST,CI,AIMIN,KF,MXX)
C --- FIND SMALLEST CELL TO ADD AND SET TEMPORARY POINTER
      REAL AVINV(MXX)
      INTEGER*2 MKG(MXX)
      DO 10 I=1,ILAST
       IF(MKG(I).EQ.0) THEN
        IF(CI+AVINV(I) .GE. AIMIN) THEN
          CI=CI+AVINV(I)
          MKG(I)=-KF
          RETURN
        ENDIF
       ENDIF
   10 CONTINUE
      WRITE(6,*) '*** ERROR IN AGG1A. NO VALUE FOUND ***'
      STOP
      END
C ***************
      SUBROUTINE AGG1B(INDX,MKG.KF,INV,Y,MXY,MXX)
C --- MAKE AGGREGATION PERMANENT
      REAL INV(MXX, MXY), Y(MXX, MXY)
      INTEGER*2 INDX(MXX),MKG(MXX)
      K=INDX(KF)
      DO 10 I=1,KF-1
       IF(MKG(I) .LT. 0) THEN
          IF(MKG(I).NE.-KF) STOP 777
          MKG(I)=KF
          L=INDX(I)
          DO 20 J=1,MXY
           INV(K,J)=INV(K,J)+INV(L,J)
             Y(K,J)=  Y(K,J)+  Y(L,J)
  20      CONTINUE
       ENDIF
  10  CONTINUE
      END
C ***************
      SUBROUTINE AGG2(AVINV,INDX,MKG,NS1,NRC,INV,Y,MXX,MXY, KF)
C --- DO ONE PASS OF AGGREGATION FROM SET S1 TO SET S0
C --- ON EACH PASS ONE ELEMENT OF S1 IS TAKEN & ADDED TO SMALLEST OF S0
      REAL INV(MXX, MXY), Y(MXX, MXY), AVINV(MXX)
```

```
      INTEGER*2 INDX(MXX),MKG(MXX)
      KF=0
C --- FIND ELEMENT OF S1  (ONLY THOSE WITH POINTER MKG(I)=0)
      DO 10 I=1,NS1
       IF(MKG(I).EQ.0) THEN
             KF=I
             GO TO 12
       ENDIF
   10 CONTINUE
   12 CONTINUE
C --- IF KF STILL 0 THEN NO MORE ELEMENTS IN S1 LEFT
      IF(KF.EQ.0) RETURN
C
C --- FIND SMALLEST ELEMENT OF S0 AND ADD TO IT. ONLY WITH MKG(I)=32767
      ISM=NRC
      SMALL=AVINV(ISM)
      DO 20 I=1, NRC
       IF(MKG(I).EQ.32767) THEN
        IF(AVINV(I).LT.SMALL) THEN
           ISM=I
           SMALL=AVINV(I)
        ENDIF
       ENDIF
   20 CONTINUE
C --- JOIN ELEMENT KF TO ELEMENT ISM
      AVINV(ISM)=AVINV(ISM) + AVINV(KF)
      MKG(KF)=ISM
      L=INDX(KF)
      K=INDX(ISM)
      DO 30 J=1,MXY
       INV(K,J)=INV(K,J)+INV(L,J)
         Y(K,J)=  Y(K,J)+  Y(L,J)
   30 CONTINUE
      END
C ***************
      SUBROUTINE CMPRS(INV,Y,MXX,MXY,NRC,NRCOLD,AIMIN,AVINV)
      REAL INV(MXX, MXY), Y(MXX, MXY), AVINV(MXX)
C --- COMPRESS INV,Y IN PLACE.  MOVE ALL ROWS GE AIMIN TO TOP
      NRCOLD=NRC
      NRC=0
      DO 10 I=1,NRCOLD
       AI=CAINV(INV,I,MXX,MXY)
       IF(AI .GE. AIMIN) THEN
C ---       TRANSFER ACTIVE CELL I ---> NRC
            NRC=NRC+1
            AVINV(NRC)=AI
            DO 20 J=1,MXY
             INV(NRC,J)=INV(I,J)
               Y(NRC,J)=  Y(I,J)
   20       CONTINUE
       ENDIF
```

```
   10  CONTINUE
       END
C ***************
       FUNCTION CAINV(INV,I,MXX,MXY)
       REAL INV(MXX, MXY)
C ---  COMPUTE AVERAGE INVENTORY FOR ROW I
        CAINV=0
        DO 10 J=1,MXY
         CAINV=CAINV+INV(I,J)
    10  CONTINUE
        CAINV=CAINV/MXY
       END
C ***************
       SUBROUTINE DSPTBL(INV,Y,AVINV,PTRTBL,INDX,AIMIN,NRC,MKG,MXX,MXY,
      *                  SYCSG,SMOS)
C ---  DISPLAY TABLE IN SORT SEQUENCE
       REAL INV(MXX, MXY), Y(MXX, MXY), AVINV(MXX)
       INTEGER*2 PTRTBL(MXX, 2), INDX(MXX),MKG(MXX)
       INTEGER SYCSG(*)
       INTEGER SMOS(*)
       INTEGER IATT(2)
       CHARACTER*1 STI
       WRITE(6,121)
       WRITE(6,*) 'INV. THRESHOLD MIN. VALUE=',AIMIN
C
       WRITE(6,*) '    I   INDX    AVG         MKG         INVENTORY/LOSSES'
       DO 200 K=1,NRC
        STI=' '
        I=INDX(K)
        AI=AVINV(K)
        IF(AI .LT. AIMIN) STI='$'
        IATT(1)=SMOS(PTRTBL(I,1))
        IATT(2)=SYCSG(PTRTBL(I,2))
        WRITE(6,122)K,I,AI,MKG(K),STI,(INV(I,J),J=1,10 ),(IATT(J),J=1,2),
      *           PTRTBL(I,1),PTRTBL(I,2)
        WRITE(6,123)                   (  Y(I,J),J=1,10 )
   200 CONTINUE
C
   121 FORMAT(///)
   122 FORMAT(/2I5,F8.3,I9,1X,A2, 10F7.2, 5X, 6I5)
   123 FORMAT( 30X, 10F7.2)
       END
C ***************
       SUBROUTINE SORT2(Y,INDX, N)
C ---  INPLACE SORT USING SHELL ALGORITHM ********
C ---  SORTS ON Y AND DOES SAME REORDERING ON INDEXES INDX
       REAL Y(N),TEMP
       INTEGER GAP
       INTEGER*2 INDX(N), ITEMP
       LOGICAL EXCH
C
```

```
      GAP=(N/2)
 5    IF (.NOT.(GAP.NE.0)) GO TO 500
10       CONTINUE
            EXCH=.TRUE.
            K=N-GAP
            DO 200 I=1,K
            KK=I+GAP
            IF(.NOT.(Y(I).GT.Y(KK))) GO TO 100
                    TEMP=Y(I)
                    Y(I)=Y(KK)
                    Y(KK)=TEMP
                    ITEMP=INDX(I)
                    INDX(I)=INDX(KK)
                    INDX(KK)=ITEMP
                    EXCH=.FALSE.
100         CONTINUE
200         CONTINUE
         IF (.NOT.(EXCH)) GO TO 10
         GAP=(GAP/2)
      GO TO 5
500   CONTINUE
      RETURN
      END
C
      SUBROUTINE BKDOWN(PTBL,NPT,PTRTBL,NRC,INDX,MKG,MXX,MXY,
     *    SINV,SY,INV,Y,BKTBL,NBK    )
C --- BREAKDOWN AGGREGATED VALUES BY THE 3RD DIMENSION SVC/CS
      REAL INV(MXX,MXY),Y(MXX,MXY). SINV(MXX,MXY),SY(MXX,MXY)
      INTEGER*2 PTRTBL(MXX, 2), INDX(MXX), MKG(MXX)
      INTEGER*2 PTBL(MXX, 3), BKTBL(MXX,3)
      REAL*8 TINV,TY
      NBK=0
C --- TRAVERSE MKG ARRAY AND BUILD BKTBL
      DO 10 I=1,NRC
       IF(MKG(I).NE.32767) THEN
                           ICELL=MKG(I)
       ELSE
                           ICELL=I
       ENDIF
       IX=INDX(I)
       IM=PTRTBL(IX,1)
       IY=PTRTBL(IX,2)
       CALL BLDBK(ICELL,IM,IY,PTBL,NPT,MXX.BKTBL,NBK)
   10 CONTINUE
C --- DISPLAY BKTBL PRIOR TO SORTING
      WRITE(6,101) (I,(BKTBL(I,J),J=1,3), I=1,NBK)
      CALL SORT3(BKTBL,NBK,MXX)
      WRITE(6,101) (I,(BKTBL(I,J),J=1,3), I=1,NBK)
C --- SUMMARIZE SINV,SY INTO INV,Y FOR MATCHING ENTRIES IN BKTBL
      CALL SUMBK(BKTBL,NBK,MXX,SINV,SY,INV,Y,MXY)
      WRITE(6,102) (I,(INV(I,J),J=1,12 ),(BKTBL(I,J),J=1,2), I=1,NBK)
```

```
      WRITE(6,102) (I,(  Y(I,J),J=1,12 ),(BKTBL(I,J),J=1,2), I=1,NBK)
  101 FORMAT(I4, 3I6)
  102 FORMAT(I4, 12F7.2,10X,2I4)
      END
C
      SUBROUTINE BLDBK(ICELL,IM,IY,PTBL,NPT,MXX,BKTBL,NBK)
      INTEGER*2 PTBL(MXX, 3), BKTBL(MXX,3)
C --- RECORD ALL ENTRIES IN PTBL WITH MATCHING IM,IY IN BKTBL
      DO 10 I=1,NPT
       IF(PTBL(I,1).EQ.IM .AND. PTBL(I,2).EQ.IY) THEN
C ---        INSTALL WITH CELL ID, IW & POINTER
             NBK=NBK+1
             BKTBL(NBK,1)=ICELL
             BKTBL(NBK,2)=PTBL(I,3)
             BKTBL(NBK,3)=I
       ENDIF
  10  CONTINUE
      END
C ***************
      SUBROUTINE SORT3(T,N,MXX)
C --- INPLACE SORT USING SHELL ALGORITHM ********
C --- SORTS ON 1ST 2 COLS. OF T & DOES SAME REORDERING ON 3RD COLUMN
      INTEGER*2 T(MXX,3), ITEMP
      INTEGER GAP
      LOGICAL EXCH
C
      GAP=(N/2)
  5   IF (GAP.EQ.0) GO TO 500
 10      CONTINUE
            EXCH=.FALSE.
            K=N-GAP
            DO 200 I=1,K
            KK=I+GAP
            IF(T(I,1).GT.T(KK,1)   .OR.
     *               (T(I,1).EQ.T(KK,1)  .AND.  T(I,2).GT.T(KK,2))  ) THEN
                  IT1=T(I,1)
                  IT2=T(I,2)
                  IT3=T(I,3)
                  T(I,1)=T(KK,1)
                  T(I,2)=T(KK,2)
                  T(I,3)=T(KK,3)
                  T(KK,1)=IT1
                  T(KK,2)=IT2
                  T(KK,3)=IT3
                  EXCH=.TRUE.
            ENDIF
 200        CONTINUE
         IF (EXCH) GO TO 10
         GAP=(GAP/2)
      GO TO 5
 500  CONTINUE
```

```
      RETURN
      END
C
      SUBROUTINE SUMBK(BKTBL,NBK,MXX,SINV,SY,INV,Y,MXY)
C --- CREATE AGGREGATED ARRAYS INV,Y FROM CELL & 3RD DIM. INFO. IN BKTBL
      REAL INV(MXX,MXY),Y(MXX,MXY), SINV(MXX,MXY),SY(MXX,MXY)
      INTEGER*2 BKTBL(MXX,3)
      REAL*8 TINV,TY
      IP=0
      I1=-1
      I2=-1
      TINV=0
      TY=0
      DO 10 I=1,NBK
       IF(BKTBL(I,1).NE.I1 .OR. BKTBL(I,2).NE.I2) THEN
C ---               CHANGE OF CELL,IW IDENTIFIERS
                    IP=IP+1
                    I1=BKTBL(I,1)
                    I2=BKTBL(I,2)
                    DO 15 J=1,MXY
                     INV(IP,J)=0
                       Y(IP,J)=0
   15               CONTINUE
                    BKTBL(IP,1)=I1
                    BKTBL(IP,2)=I2
       ENDIF
C ---  ACCUMULATE
       I3=BKTBL(I,3)
       DO 20 J=1,MXY
        INV(IP,J)=INV(IP,J)+SINV(I3,J)
          Y(IP,J)=  Y(IP,J)+  SY(I3,J)
        TINV=TINV+SINV(I3,J)
          TY=  TY+  SY(I3,J)
   20  CONTINUE
   10 CONTINUE
C
      NBK=IP
      WRITE(6,*) '==== TOTAL INV,Y AFTER BREAKDOWN=',TINV,TY
      END
```

F. FORTRAN PROGRAM: WSEAS

```
      PROGRAM    WSEAS
******************************************************************
*     WINTERS THREE PARAMETER FORECASTING MODEL                  *
*                                                                *
*     VARIABLES USED                                             *
*          N = NUMBER OF SEASONS = 4                             *
```

```
*           TMAX = PERIODS OF AVAILABLE DATA                       *
*           ALPHA= SMOOTHING CONSTANT                              *
*           BETA = SMOOTHING CONSTANT                              *
*           GAMMA= SMOOTHING CONSTANT                              *
*           D()  = CURRENT ATTRITION IN PERIOD T                   *
*           DF() = FORECAST ATTRITION IN PERIOD T                  *
*           EOF()= TRANSFORMED ERROR OF FORECAST                   *
*           MAD()= MEAN ABSOLUTE DEVIATION ON FORECAST             *
*                                                                  *
********************************************************************

*     Input Initialization

      INTEGER N, TMAX
      PARAMETER (N = 4, TMAX = 48)
*                           /*48=Qtrs on Tape*/

      INTEGER I, K, L, M, KK, AL, BET, GAM
      REAL ALPHA, BETA, GAMMA, INDEX(-3:TMAX), SMOOTH(0:TMAX),
     +     TREND(0:TMAX), INV(TMAX), LOS(TMAX), DINV(5), DLOS(5),
     +     A(N), F(N), EOF(N), D(TMAX), DF(TMAX+4), MAD(4), MSE(4)

*  // Initialize values smoothing constants //
      ALPHA = 0.45
      BETA  = 0.35
      GAMMA = 0.10

*  // DO LOOP to Run Validation on each Rank of Current MOS Group //
      DO 1 KK = 1,5

*  // Bootstrap INDEX and TREND to initiate Seasonality flow //
      DO 5 I = 5,8
        INDEX(I) = 1.00
    5 CONTINUE
      TREND(8) = 0.01
      DO 6 I = 1,4
        MAD(I) = 0.0
        MSE(I) = 0.0
    6 CONTINUE

*  // Read Data; Must have min INV() = 1; Compute Attrition Rate //
      DO 10 I = 1,TMAX
        READ (11,101) (DLOS(J), J=1,5)
        READ (13,101) (DINV(J), J=1,5)
        LOS(I) = DLOS(KK)
        INV(I) = DINV(KK)
        IF (INV(I) .LT. 1.0) INV(I) = 1.0
        D(I)   = LOS(I) / INV(I)
        IF (D(I) .GE. 1.0) D(I) = (LOS(I)+1) / (INV(I)+2)
   10 CONTINUE
```

```
      REWIND(11)
      REWIND(13)
  101 FORMAT (5F10.0)

*   // Winters Forecast Computations //
      SMOOTH(8) = D(9)
      DO 15 I = 9, TMAX
        SMOOTH(I) = (ALPHA * D(I) / INDEX(I-4)) + (1-ALPHA) *
     +                (SMOOTH(I-1) + TREND(I-1))
        TREND(I)  = GAMMA * (SMOOTH(I) - SMOOTH(I-1)) + (1-GAMMA) *
     +                TREND(I-1)
        INDEX(I)  = (BETA * D(I) / SMOOTH(I)) + (1-BETA) * INDEX(I-4)
*       INDEX(I)  = INDEX(I - 1)
        DF(I+1) = (SMOOTH(I) + TREND(I)) * INDEX(I-3)

*   // Compute EOF and Sum the MAD //
        IF ((I .GE. 9) .AND. (I .LE. 38)) THEN
          F(1) = DF(I+1)
          A(1) = D(I+1)
          DO 20 M=2,N
            DF(I+M) = (SMOOTH(I) + M * TREND(I)) * INDEX(I+M-N)
            F(M) = DF(I+M)
            A(M) = D(I+M)
   20     CONTINUE
          DO 25 M = 1,4
            IF (F(M) .GE. 1.0000) F(M) = 0.9999
            IF (F(M) .LT. 0.0001) F(M) = 0.0001
            EOF(M) = (A(M)-F(M)) * (INV(I+M))**.5
            MAD(M) = MAD(M) + ABS(EOF(M)/30)
            MSE(M) = MSE(M) + ((EOF(M)**2)/30)
   25     CONTINUE
        ENDIF
   15 CONTINUE

      WRITE(14,125) (MAD(M), M=1,4)
  125 FORMAT(2X,4(F12.6))

      WRITE(15,126) (MSE(M), M=1,4)
  126 FORMAT(2X,4(F12.6))

    1 CONTINUE
      STOP
      END
```

G. FORTRAN PROGRAM: HSSEAS

```
      PROGRAM     HSSEAS
***************************************************************************
```

```
*     HARRISON STEVENS: SHORT TERM FORECASTING MODEL                *
*     ***  MULTIPLICATIVE SEASONALITY ON CONTINUATION  ***          *
*                                                                   *
*     VARIABLES USED                                                *
*          N = NUMBER OF STATES = 4                                 *
*          PI(J) = PROB OF STATE J; J = 1,2,...,N                   *
*          RE() = RATIO PARAMETER ON OBSERVATIONAL NOISE            *
*          RG() = RATIO PARAMETER ON TREND PERTERBATION             *
*          RD() = RATIO PARAMETER ON SLOPE PERTERBATION             *
*          VO   = VARIANCE LAW                                      *
*          D()  = CURRENT CONTINUATION RATE FOR PERIOD T            *
*          VCE  = VARIANCE IN OBSERVATIONAL NOISE (CURRENT)         *
*          VD   = VARIANCE IN SLOPE PERTURBATION                    *
*          VG   = VARIANCE IN TREND PERTURBATION                    *
*          MC   = CURRENT TREND VALUE                               *
*          BC   = CURRENT SLOPE VALUE                               *
*          VCMM = CURRENT COV MATRIX ELEMENT (1,1)                  *
*          VCMB = CURRENT COV MATRIX ELEMENT (1,2) AND (2,1)        *
*          VCBB = CURRENT COV MATRIX ELEMENT (2,2)                  *
*          S()  = SEASONAL VALUES                                   *
*          QTR  = PRESENT SEASON                                    *
*          Q()  = UPDATED STATE PROBABILITY                         *
*          R11()= SUM OF VAR/COV MATRIX ELEMENT (1,1)               *
*          R12()= SUM OF VAR/COV MATRIX ELEMENT (1,2), (2,1)        *
*          R22()= SUM OF VAR/COV MATRIX ELEMENT (1,1)               *
*          VE() = EXPECTED OBSERVATIONAL NOISE                      *
*          M()  = EXPECTED TREND VALUE                              *
*          B()  = EXPECTED SLOPE VALUE                              *
*          A1() = SMOOTHING CONSTANT                                *
*          A2() = SMOOTHING CONSTANT                                *
*          VMM()= NEXT COV MATRIX ELEMENT (1,1)                     *
*          VMB()= NEXT COV MATRIX ELEMENT (1,2) AND (2,1)           *
*          VBB()= NEXT COV MATRIX ELEMENT (2,2)                     *
*          DF() = FORECAST CONTINUATION RATE                        *
*          EOF  = ERROR OF FORECAST (TRANSFORMED)                   *
*          MAD  = MEAN ABSOLUTE DEVIATION OF FORECAST               *
*                                                                   *
*********************************************************************


*     Input Initialization

      INTEGER N, TMAX
      PARAMETER (N = 4, TMAX = 48)
                              /*48=Qtrs on Tape*/

      INTEGER I, J, T, QTR, REPLY, KK, KKY
      REAL*8 RE(N), RG(N), RD(N), PI(N), D(TMAX), LOS(TMAX), MSE(4),
     +   S(4), INV(TMAX), VCE(N), VG(N), VD(N), MC(N), BC(N), VCMM(N),
     +   VCMB(N), VCBB(N), Q(N), ET(N), CNST, R11(N,N), R12(N,N), V[illegible]
     +   R22(N,N), VE(N,N), A1(N,N), A2(N,N), MAD(4), M(N,N), B(N,N),
```

```
     +    VMM(N,N), VMB(N,N), VBB(N,N), P(N,N), EH(TMAX), DF(TMAX+1),
     +    A(N), F(N), EOF(N), DENOM, SOLD(N), BETA, DLOS(5), DINV(5)

      V0 = .001
      BETA = .35

    5 CALL EXCMS('FILEDEF 10 DISK VARIABLE DATA A')

*   // Read in Harrison Stevens Parameters //
      DO 10 J=1,N
        READ(10,*) PI(J), RE(J), RC(J), RD(J)
   10 CONTINUE

*   // Loop To Perform Forecast for each of 5 Ranks in this MOS Group //
      DO 1 KK=1,5

*   // Read in Data from GRP* MAT Files; Do Not Let INV() = 0 //
      DO 8 I = 1,TMAX
        READ (11,201) (DLOS(J), J=1,5)
        READ (13,201) (DINV(J), J=1,5)
      LOS(I) = DLOS(KK)
      INV(I) = DINV(KK)
      IF (INV(I) .LT. 1.0) INV(I) = 1 0
    8 CONTINUE
  201 FORMAT (5F10.0)
      REWIND(10)
      REWIND(11)
      REWIND(13)


*   // Compute Continuation Rates; Do not allow D() - 0  //
      DO 11 I = 1,TMAX
        D(I) = (INV(I)-LOS(I)) / INV(I)
        IF (D(I) .LT. .00001) D(I) = (INV(I)-LOS(I)+1)/(INV(I)+2)
   11 CONTINUE

      DO 12 I = 1,4
        MAD(I) = 0.0
        MSE(I) = 0.0
   12 CONTINUE
      T = 3

*   // Compute Initial Seasonal Values based upon Continuation //
      DENOM = 1
      DO 14 K=9,12
        DENOM = DENOM * D(K)
   14 CONTINUE
      DO 15 K=1,4
        S(K) = (D(K+8)) / (DENOM**.25)
   15 CONTINUE
```

```
*  // Flag a Conditional //
      DO 20 J=1,N
        IF (RE(N) + RG(N) + RD(N) + 0.0001 .LT. RE(J) + RG(J) +
     +   RD(J)) THEN
          WRITE(*,*) 'Error Statement re: Parameters'
          GOTO 1000
        ENDIF

*  // Define Variances in terms of Ratios of the Basic Noise V0 //
        VCE(J) = RE(J) * V0
        VG(J)  = RG(J) * V0
        VD(J)  = RD(J) * V0

*  // Initialize Values for the Condensed Parameters //
        MC(J) = D(J) / S(J)
        BC(J) = 0.0
        VCMM(J) = 0.0
        VCMB(J) = 0.0
        VCBB(J) = 0.0
        Q(J) = PI(J)
   20 CONTINUE


*  // Start Iterative Algorithm //
  999 CONTINUE

*  // Set Proper Season //
      QTR = MOD(T,4)
      IF (QTR .EQ. 0) QTR = 4

      DO 40 I=1,N

*  // Check to Prevent Computer Precision Error, ET ---> ZERO //
        IF (ABS(D(T)-((MC(I)+BC(I))*S(QTR)+.00001)) .LT. .00001) THEN
          ET(I) = 0
        ELSE
          ET(I) = D(T) - (MC(I) + BC(I)) * S(QTR)
        ENDIF
   40 CONTINUE

*  // for Summing, set CNST = 0 //
      CNST = 0.0

      DO 60 I=1,N
        DO 50 J=1,N
          R11(I,J) = VCMM(I) + 2.0 * VCMB(I) + VCBB(I)
     +                         + VG(J) + VD(J)
          R12(I,J) = VCMB(I) + VCBB(I) + VD(J)
          R22(I,J) = VCBB(I) + VD(J)
          VE(I,J) = (S(QTR)**2) * R11(I,J) + VCE(J)
          A1(I,J) = S(QTR) * R11(I,J) / VE(I,J)
```

```
          A2(I,J) = S(Q,R) * R12(I,J) / VE(I,J)

*   // Joint Posterior Distribution at Time t //
          M(I,J)   = MC(I) + BC(I) + A1(I,J) * ET(I)
          B(I,J)   = BC(I) + A2(I,J) * ET(I)
          VMM(I,J) = R11(I,J) - (A1(I,J)**2) * VE(I,J)
          VMB(I,J) = R12(I,J) - A1(I,J) * A2(I,J) * VE(I,J)
          VBB(I,J) = R22(I,J) - (A2(I,J)**2) * VE(I,J)

*   // Develop the State Transitional Matrix //

*   // Check to Prevent Computer Precision Error, P --> ZERO  //
          IF ((ET(I)**2/(2*VE(I,J))) .GT. 50.0) THEN
            P(I,J) = 0.0
          ELSE
            P(I,J)=Q(I) * PI(J) * EXP((-(ET(I)**2))/(2 * VE(I,J)))
     +                / SQRT(6.28318 * VE(I,J))
          ENDIF

          CNST = CNST + P(I,J)

   50   CONTINUE
   60 CONTINUE

*   // P(I,J) scale change //
      DO 80 I=1,N
        DO 70 J=1,N
          P(I,J) = P(I,J) / CNST
   70   CONTINUE
   80 CONTINUE

*   // Perform the Condensation Step //
      DF(T+1) = 0.0
      DO 120 J=1,N
        Q(J)   = 0.0
        MC(J)  = 0.0
        BC(J)  = 0.0
        VCMM(J) = 0.0
        VCMB(J) = 0.0
        VCBB(J) = 0.0
        DO 90 I=1,N
          Q(J) = Q(J) + P(I,J)
   90   CONTINUE

        DO 100 I=1,N
          MC(J) = MC(J) + P(I,J) * M(I,J) / Q(J)
          BC(J) = BC(J) + P(I,J) * B(I,J) / Q(J)
  100   CONTINUE

*   // Develop the Variance-Covariance of Condensed Values //
        DO 110 I=1,N
```

```
          VCMM(J) = VCMM(J) + P(I,J) * (VMM(I,J) + (((M(I,J)
     +              - MC(J))**2) / Q(J)))
          VCMB(J) = VCMB(J) + P(I,J) * (VMB(I,J) + ((M(I,J)
     +              - MC(J)) * (B(I,J) - BC(J)) / Q(J)))
          VCBB(J) = VCBB(J) + P(I,J) * (VBB(I,J) + (((B(I,J)
     +              - BC(J))**2) / Q(J)))
  110   CONTINUE

*  // Compute the forecast for time T+1 //
        IF (QTR .EQ. 4) QTR = 0
        DF(T+1) = DF(T+1) + (Q(J) * (MC(J) + BC(J)) * S(QTR+1))
  120 CONTINUE

*  //  Compute Error of Forecast out next four periods //
      IF ((T .GE. 9) .AND. (T .LE. 38)) THEN
        F(1) = DF(T+1)
        A(1) = D(T+1)
        DO 122 K=2,4
          A(K) = D(T+K)
          IF (QTR+K .LE. 4) THEN
            F(K) = F(1) * S(QTR+K)
          ELSE
            F(K) = F(1) * S(QTR-4+K)
          ENDIF
  122   CONTINUE
        DO 123 K = 1,4

*  //  Prevent Divide by zero  //
          IF (F(K) .GE. 1.0000) F(K) = 0.9999
          IF (F(K) .LT. 0.00001) F(K) = 0.00001
          EOF(K) = (A(K)-F(K)) * (INV(T+K))**.5

*  // Sum to Compute the MAD and MSE //
          MAD(K) = MAD(K) + ABS(EOF(K)/30)
          MSE(K) = MSE(K) + (EOF(K)**2)/30
  123   CONTINUE
        WRITE(12,124) (EOF(K), K=1,4)
  124   FORMAT(2X,4(F12.5))
      ENDIF
      IF (QTR .EQ. 0) QTR = 4

*  // Check Stopping Rule //
      IF (T .LT. TMAX) THEN

*  // Record the old Seasonal Values //
        DO 127 K = 1,4
          SOLD(K) = S(K)
  127   CONTINUE

*  // Update New Seasonal Values based upon Continuation//
        DENOM = 1
```

```
      DO 125 K=1,4
        IF ((T+K-4) .GT. 0) THEN
          DENOM = DENOM * D(T+K-4)
        ELSE
          DENOM = DENOM**2
        ENDIF
125   CONTINUE
      DO 126 K=1,4
        IF ((T+K-4) .LT. 1) GOTO 126
        IF ((QTR+K) .LE. 4) THEN
          S(QTR+K) = (D(T+K-4)) / (DENOM**.25)
        ELSE
          S(K+QTR-4) = (D(T+K-4)) / (DENOM**.25)
        ENDIF
126   CONTINUE

*  //  Complete Weighted Update of Seasonal Values  //
      DO 128 K = 1,4
        S(K) = (S(K)**BETA) * (SOLD(K)**(1-BETA))
128   CONTINUE

      T = T + 1

*  // **Continue Iterative Process** //

      GOTO 999
     ENDIF

1000 CONTINUE

     WRITE(14,1001) (MAD(K), K=1,4)
1001 FORMAT(2X,4(F12.5))

     WRITE(15,1002) (MSE(K), K=1,4)
1002 FORMAT(2X,4(F12.5))
   1 CONTINUE

     STOP
     END
```

APPENDIX C

TABLE 1. MOS GROUPS

Group Name	MOS's	Sm MOS Grp	Lrg MOS Grp	Mjr MOS Grp
Combat	0302	1	1	1
Combat Support	0802 1302 1802 1803	2		
Combat Service 1	0108 0202 2502 2602	3	2	
Combat Service 2	3415 4002 4302 5803	4		
Combat Logistic	0402 3002 3060 3502 6002	5		
Air Control	7204 7208 7210 7320	6		
Fixed Wing Pilot	7501 7511 7522 7542 7543 7545 7576	7	3	2
F-18 Pilot	7521 7523	8		
Rotary Wing Pilot +	7556 7557 7562 7563 7564 7565 7566	9	4	
NFO +	7508 7509 7581 7583 7584 7585 7586 7587 7588	10		
Basic Ground	0101 0201 0301 0401 0801 1301 1801 2501 2601 3001 3401 3501 4001 4301 4401 5801 6001 7201 7301 9901	11	5	3
Student Aviator	7580 7597 7598 7599	12		
Basic Pilot	7500 7510 7520 7540 7550 7560 7575	13		
Lawyer	4402	14	6	4

TABLE 2. YCS EXPANSION BOUNDS

MOS Groups	Small MOS Groups	YCS Group Bounds
Fixed Wing Pilots, Lawyers	7, 8, 14	(1-6,8-19) (7) (20-25) (26)
Rotary Wing Pilots, Naval Flight Officers	9, 10	(1-5,8-19) (6,7) (20-25) (26)
All Others	1-6, 11-13	(1-3,6-19) (4,5) ([illegible]0-25) (26)

LIST OF REFERENCES

Amir Elseranegy, H CART Program: Implementation of the CART Program and It's Application to Estimating Attrition Rates, Master's Thesis, Naval Postgraduate School, Monterey, CA, December 1985.

Decision Systems Associates, Inc., Functional Description for the Development of the Officer Planing and Utilization System (OPUS), Rockville, MD, 1986.

Dickinson, C. R., Refinement and Extension of Shrinkage Techniques in Loss Rate Estimation of Marine Corps Officer Manpower Models, Master's Thesis, Naval Postgraduate School, Monterey, CA, March 1988.

Harrison, P. J. and Scott, F. A., A Development System for Use in Short Term Sales Forecasting Investigations, Paper presented at the Operations Research Society Annual Conference, Shrivenham, England, 1965.

Harrison, P. J., Exponential Smoothing and Short Term Sales Forecasting, Management Science, Volume 30, Number 11, 1967.

Harrison, P. J. and Stevens, C. F., A Bayesian Approach to Short Term Forecasting, Operations Research Quarterly, Volume 22, Number 4, 1971.

Hogan, D. L. Jr., The Use of Exponential Smoothing to Produce Yearly Updates of Loss Rate Estimates in Marine Corps Manpower Models, Master's Thesis, Naval Postgraduate School, Monterey, CA, June 1986.

Larsen, R. W., The Aggregation of Population Groups to Improve the Predictability of Marine Corps Officer Attrition Estimation, Master's Thesis, Naval Postgraduate School, Monterey, CA, December 1987.

Makridakis, S. and Wheelwright, S. C., Interactive Forecasting, Holden-Day, Inc., San Francisco, CA, 1978.

Misiewicz, J. M., Extension of Aggregation and Shrinkage Techniques Used in the Estimation of Marine Corps Officer Attrition Rates, Master's Thesis, Naval Postgraduate School, Monterey, CA, September 1989.

Read, R. R., Naval Postgraduate School Report NPS55-88-006, The Use of Shrinkage Techniques in the Estimation of Attrition Rates for Large Scale Manpower Models, 27 July 1988.

Robinson, J. R., Limited Translation Shrinkage Estimation of Loss Rates in Marine Corps Manpower Models, Master's Thesis, Naval Postgraduate School, Monterey, CA, March 1986.

Siegel, B., Methods for Forecasting Officer Loss Rates, Navy Personnel Research and Development Center, San Diego, CA, 1983.

Trigg, D. W. and Leach, A. G., Exponential Smoothing With an Adaptive Response Rate, Operational Research Quarterly, Volume 18, Number 1, 1967.

Tucker, D. D., Loss Rate Estimation in Marine Corps Officer Manpower Models, Master's Thesis, Naval Postgraduate School, Monterey, CA, September 1985.

Washburn, Alan, A Short Introduction to Kalman Filters, Naval Postgraduate School, Monterey, CA.

Yacin, N., Application of Logistic Regression to the Estimation of Manpower Attrition Rates, Master's Thesis, Naval Postgraduate School, Monterey, CA, March 1987.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Professor Robert R. Read, (Code OR/Re) Naval Postgraduate School Monterey, CA 93943-5000	2
4.	Professor Lyn R. Whitaker, (Code OR/Wh) Naval Postgraduate School Monterey, CA 93943-5000	1
5.	Major Charles J. Mehalic MAGTF Warfighting Center (WF13) Marine Corps Combat Development Center Quantico, VA 22134-5001	2
6.	Mr. Rod Morton Decision Systems Associates, Inc. 350 Fortune Terrace Rockville, MD 20854-2995	1
7.	Commandant of the Marine Corps Code MI (Attn: Captain Lon Yeary) Headquarters, U.S. Marine Corps Washington, D.C. 20380-0001	1
8.	Commandant of the Marine Corps Code MPP-31 (Attn: Captain Thomas Finn) Headquarters, U.S. Marine Corps Washington, D.C. 20380-0001	1
9.	Major John M. Misiewicz MAGTF Warfighting Center (WF13) Marine Corps Combat Development Center Quantico, VA 22134-5001	1

10. Director 1
    Center for Naval Analysis
    4401 Ford Avenue
    Alexandria, VA 22302

11. Commandant of the Marine Corps 1
    Code TE06
    Headquarters, U.S. Marine Corps
    Washington, D.C. 20380-0001